NIST-GCR-94-662

TRANSIENT COOLING OF A HOT SURFACE BY DROPLETS EVAPORATION

G. White, S. Tinker,
M. di Marzo
Mechanical Engineering Department
University of Maryland
College Park, MD 20742

December 1993
Issued November 1994

U.S. Department of Commerce
Ronald H. Brown, *Secretary*
Technology Administration
Mary L. Good, *Under Secretary for Technology*
National Institute of Standards and Technology
Arati Prabhakar, *Director*

Notice

This report was prepared for the Building and Fire Research Laboratory of the National Institute of Standards and Technology under grant number 70NANB1H1173. The statement and conclusions contained in this report are those of the authors and do not necessarily reflect the views of the National Institute of Standards and Technology or the Building and Fire Research Laboratory.

TRANSIENT COOLING OF A HOT SURFACE BY DROPLETS EVAPORATION

Final Report

G. White, S. Tinker, M. di Marzo

Prepared for the
Building and Fire Research Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899

Mechanical Engineering Department
University of Maryland
College Park, MD 20742

December 1993

ABSTRACT

A computer code is developed and tested which simulates the transient evaporation of a single liquid droplet from the surface of a semi-infinite solid subject to radiant heat input from above. For relatively low temperature incident radiation, it is shown that the direct absorption of radiant energy by the droplet can be treated as purely boundary conditions, while a model for higher temperature incident radiation would require the addition of constant heat source terms. The heat equation is numerically coupled between the liquid and solid domains by using a predictor-corrector scheme. Three one-dimensional solution schemes are used within the droplet: a start-up semi-infinite medium solution, a tridiagonal Crank-Nicholson transient solution, and a steady-state solution. The solid surface temperatures at each time step are calculated through careful numerical integration of an axisymmetric Green's functions solution equation with the forcing function given by the past lower droplet surface and solid-vapor boundary heat fluxes. The time step is increased after a sensitive initial period to allow for reasonable run times. Two geometry models are included which give the droplet height as a function of current droplet volume and initial wetted radius; the second allows inclusion of the effects of initial contact angle and receding angle. Using water as the liquid and Macor, a low-thermal conductivity material, as the solid, the program output was compared to the experimental results in this line of research. They correlate well to the experiments in which the critical geometric shape factor and evaporation time were most easily measured.

FOREWORD

This report describes the research performed during the period September 1992 - August 1993 under a joint research program between the Mechanical Engineering Department of the University of Maryland at College Park and the Building and Fire Research Laboratory of the National Institute of Standards and Technology. The research was conducted in the laboratories of the BFRL by G. White and S. Tinker, Graduate Research Assistants of the ME Department under the joint supervision of Dr. M. di Marzo (ME Department - UMCP) and Dr. D. Evans (BFRL - NIST). This report also constitutes the Master Thesis of Mr. G. White, which has been completed and will be defended in the month of December 1993. Ms. S. Thinker performed experiments on dropwise evaporation which will be included in the final report for next grant period. She was also responsible for the formulation of a two-parameters model for the description of the droplet shape which was used by G.White in his thesis. Ms. Tinker is currently working on the formulation and validation of a multi-droplet model for the prediction of sparse spray evaporative cooling.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

a	droplet apex
A, B	linearization constants for liquid-vapor boundary condition (m^{-1}, K/m)
$c_{p,air}$	specific heat at constant pressure of air
D	mass diffusivity of steam in air
$E_{\lambda,b}$	blackbody spectral emissive power
$E_{\lambda,s}$	radiative source spectral emissive power
$E_{\lambda,\theta}$	incident radiative flux at polar angle θ
f_θ	fractional coverage at polar angle θ
F	radiative heat flux inside droplet
F	radiative heater coils to tile view factor
$F_{\lambda,\theta}$	radiative flux inside droplet at depth z due to energy from θ at wavelength λ
F_ε	radiative heater coils to horizontal droplet surface view factor
h	convective heat transfer coefficient
h_m	convective mass transfer coefficient
h_o	overall heat transfer coefficient (= $h + h_r$)
h_r	radiative heat transfer coefficient of solid
H	volumetric heat generation inside droplet due to radiation absorption
H_c	residual volumetric heat generation inside droplet (assumed constant)
k	thermal conductivity of liquid
k_s	thermal conductivity of solid
Le	Lewis number (= α / D)
M	molecular mass
n	number of elements in annular column of water
n	index of refraction of water

N	total molar flux
N_{air}	molar flux of air
N_l	molar flux of droplet vapor
p_{sat}	saturation pressure of water vapor
p_a	ambient atmospheric pressure
q_0	initial heat flux in solid
q_{disk}	constant flux disk model flux
q_i	liquid-vapor interfacial flux
q_S	solid surface heat flux
r	radial position
r_0	radial position (dummy BEM integration variable)
R	current wetted area radius
R_0	initial wetted area radius
t	time
Δt	time step
t_0	time (dummy BEM integration variable)
t_{SIS}	time at which heat wave reaches liquid-vapor interface
t'	recollection time
T	solid temperature
T_0	initial solid temperature
T_a	ambient temperature
T_C	semi-infinite solution contact temperature
T_i	liquid-vapor interface temperature
T_L	initial liquid temperature
T_R	radiative heater coils temperature
T_S	solid surface temperature
T_{S0}	initial solid surface temperature

u temperature depression

v_{fg} volume change associated with phase change from liquid to vapor

V current droplet volume

V_0 initial droplet volume

W BEM weight matrix

x_a ambient mole fraction of droplet vapor

x_i mole fraction of droplet vapor at liquid-vapor interface

z axial system coordinate or distance below liquid-vapor surface

GREEK

α thermal diffusivity

β geometric splat factor (= R/R_s)

β Green's function dummy integration variable

γ Crank-Nicholson constant

δ height of droplet at radial position r

ε emissivity of solid surface

η nonhomogeneous volumetric heat generation term (°C)

κ_λ droplet spectral absorption coefficient

Λ latent heat of vaporization of liquid

μ direction cosine to z-axis inside droplet

ρ droplet density

ρ_θ droplet surface reflectivity

σ Stefan-Boltzmann radiation constant (= 5.670E-08 $W/m^2\text{-}K^4$)

τ evaporation time

θ laboratory polar angle

θ droplet-solid contact angle

θ_0 initial droplet-solid contact angle (geometry Model B only)

θ_r droplet-solid receding angle (geometry Model B only)

1. INTRODUCTION

Because of the very high latent heat of vaporization of water, cooling using water droplets has many engineering applications. These uses vary from the quenching of metals and cooling of turbine blades to environmental control systems. Therefore, cooling of hot surfaces has been the subject of numerous investigations. Toda [1] and Bonacina [2] designed spray cooling experiments while more recent work by Inada [3] and Makino [4] studied the evaporation of single droplets heated from below by conduction.

The theoretical transient single-droplet evaporative cooling computer code developed by the author was completed under a joint research program between the Mechanical Engineering Department of the University of Maryland and the Center for Fire Research of the National Institute of Standards and Technology. This research has focused on extending the current body of knowledge on evaporative cooling to the fire safety field. Previous experimental work [5–9] has investigated the temperature response of the surface of a semi-infinite solid surrounding a gently deposited water droplet. The surface may be heated from below by conduction or from above by radiation, but previous theoretical work [8, 10–12] has focused on the conduction case. More recent work has focused on measuring [13] or predicting [14] the surface temperatures of a solid impacted by a random droplet array. The time for reaching the surface-averaged steady-state temperature for a surface subject to a sudden droplet array is critical to the protection of equipment exposed to radiant heat. Therefore, the results of the single-droplet code presented here will be used in a multi-droplet code. To reach that goal, the code documented here is designed to output transient

solid surface temperatures and droplet evaporation times for water droplets of variable size and initial wetted area gently deposited on low-thermal conductivity semi-infinite solids subject to radiant energy.

1.1 Problem Description

The computer code documented here models the specific experimental geometry of Dawson [13] as shown in Figure 1. Heating is from above by three radiant panels operating in a temperature range of from 475–650°C. The droplet is gently deposited at the center of a semi-infinite solid Macor tile, a glass-like ceramic resistant to thermal stresses and chosen for its low thermal conductivity (1.29 W/m-K) and high emissivity of 0.84 [8]. The temperature profile along a line passing through the droplet center was measured using infrared thermography data recorded on video tape. Only initial surface temperatures less than that required for nucleate boiling in the droplet were allowed (i.e., $T_{S0} < 162$°C). Therefore, the droplet vaporization process is exclusively evaporative.

The droplet-solid geometry is shown in Figure 2. An overall energy balance predicts the amount of energy required to evaporate the droplet but cannot predict the relative proportions of the heat transfer mechanisms nor the evaporation time. Energy enters the droplet by conduction from below and radiation from above and leaves the droplet with the mass evaporated at the upper surface. A small amount of energy also leaves the upper surface by natural convection and is quantified using the experimentally measured value of h (approximately 10 W/m-K). The Chilton-Colburn analogy will be used to develop a relationship for the energy leaving by evaporation. Chapter 2 is devoted to quantifying the energy added to the droplet by

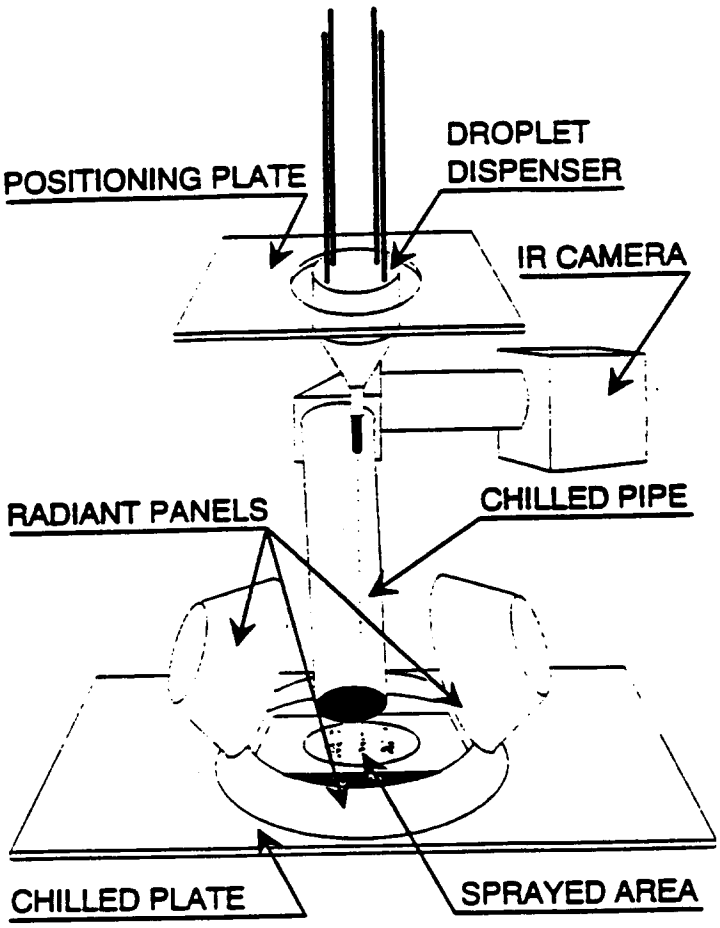

FIGURE 1
Infrared Thermography Experimental Setup

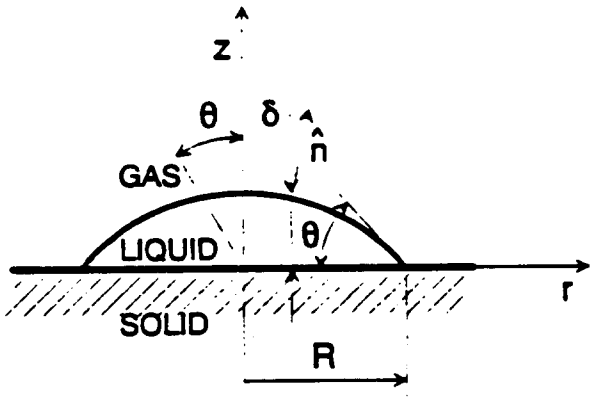

FIGURE 2
Coordinate System and Geometrical Parameters

radiation. It is shown that the radiation may be assumed to be input exclusively at the droplet boundaries. The conductive contribution from below, responsible for the surface cooling effect, requires coupling of the heat equation in the liquid and solid domains. This is accomplished using a simple but effective predict-correct method. Surface-tension-induced motion in the droplet can be neglected because of the very small induced velocities, and Rayleigh-instability-induced flow is not of concern because of the large response times. Then, the thin film assumption that conductive heat transfer in the droplet is one-dimensional in the axial direction is made based on previous results [15] for the case of heating by conduction from below. This assumption is true everywhere and at all times except near the droplet edge where the radial flux component may be as large as 10% of the total flux. The radiation case droplet tends to be thinner than the conduction case one, so the assumption can safely be applied to the present problem. The solid domain is best handled by a Green's function Boundary Element Method (BEM).

1.2 Key Assumptions

The major assumptions used by the computer code are listed in the following:

(1) Liquid flow and thus convection inside the droplet are negligible.

(2) Conduction in the droplet is one-dimensional in the axial direction.

(3) A semi-infinite solid solution can sufficiently describe the early solid-liquid interfacial fluxes.

(4) The direct radiation absorption by the droplet can be sufficiently described by radiation absorption at the boundaries only.

(5) The droplet shape may be empirically described by geometry Model A or B (subsection 3.9): spherical segment shape or pancake shape with possible shrinkage of the wetted area.

2. SECTION I - RADIATIVE HEAT INPUT MODEL

The evaporative process cools a surface by conducting sensible energy from the bulk of the solid in order to supply latent heat of vaporization. However, incident radiative heat flux (e.g. from a nearby fire) can also supply the needed energy and thus limit the cooling effect. The manner in which the radiative flux enters the system must affect the transient characteristics of the evaporative process. For example, radiation absorbed right at the droplet surface would affect the temperature-sensitive evaporative heat flux more strongly than radiation absorbed directly by the solid at the solid-liquid interface. Also radiation incident to the droplet at shallow angles tend to be reflected away from the solid entirely. Therefore, this section presents a radiative heat input model of enough detail so as to specify locations at which the radiation enters the droplet-solid system but not so complex as to make the problem extremely computationally intensive. This model is necessary for the program developed in Chapter 3, which simulates the controlled laboratory environment, because the overall energy balance provided by the known initial heat flux in the tile only gives the magnitude of the radiation absorbed by the tile but not the directions from which the radiation is incident.

2.1 Derivation of Model Equation

Because of the assumed azimuthal symmetry of the droplet and semi-infinite solid, the polar angle distribution of the incident radiation and not the azimuthal distribution is needed. Then, flux incident from polar angle θ with wavelength λ can be designated $E_{\lambda,\theta}$. If one makes the assumption that the liquid-vapor interface is horizontal and flat, then the radiative flux is only a

function of the depth z below the interface and not also of the radial position and current droplet shape. The droplet surface normal need not be calculated as a function of radial position and time. This assumption may be reasonable because the droplet is usually rather thin. Two additional assumptions are also made now: radiation scattering within the droplet is negligible and radiation reaching the liquid-solid interface is completely absorbed by the solid. Then with the effects of projected area, surface reflection, and absorption inside the droplet, the heat flux at a depth z below the droplet surface due to $E_{\lambda,\theta}$ is

$$F_{\lambda,\theta}(z) = E_{\lambda,\theta} \cos\theta \, (1 - \rho_\theta) \exp(-\kappa_\lambda z/\mu) \tag{2.1}$$

where κ_λ is the spectral absorption coefficient of the liquid, μ is the direction cosine to the z-axis, and ρ_θ is the reflectivity of the liquid surface for an angle θ between the incident radiation and the droplet normal. κ_λ acts to exponentially damp the radiative flux over the distance it has traveled inside the droplet, z/μ. μ is found from Snell's Law (neglecting the slight dependence of n on wavelength):

$$n \sin\theta' = \sin\theta; \qquad \mu = \cos\theta'; \qquad n \approx 1.33$$
$$\mu = \cos\left[\sin^{-1}(0.75 \sin\theta)\right] \tag{2.2}$$

$E_{\lambda,\theta}$ is found using $I_{\lambda,s}$, the spectral intensity at polar angle θ due to the radiative source, and the differential solid angle $d\omega$:

$$E_{\lambda,\theta} = I_{\lambda,s} \, d\omega \qquad \text{where} \quad d\omega = dA_n / R^2 \tag{2.3}$$

dA_n only covers a fraction of the hemisphere at θ, that fraction of the hemisphere occupied by the source of the radiation. The author designates this fraction the fractional coverage with symbol f_θ, the f chosen because of the relation to the view factor nomenclature. f_θ will be found for the specific laboratory geometry using analytic geometry.

Then $d\omega = 2\,\pi\, f_\theta \sin\theta\, d\theta$, and using the identity $E = \pi I$ [16], we have

$$F_{\lambda,\theta}(z) = 2\, E_{\lambda,s}\, f_\theta \cos\theta \sin\theta\, (1 - \rho_\theta) \exp(-\kappa_\lambda z/\mu)\, d\theta \qquad (2.4)$$

Integrating over all possible wavelengths and polar angles gives

$$F(z) = 2 \int_0^\infty E_{\lambda,s} \int_0^{\pi/2} f_\theta \cos\theta \sin\theta\, (1 - \rho_\theta) \exp(-\kappa_\lambda z/\mu)\, d\theta\, d\lambda \qquad (2.5)$$

The volumetric heat generation due to absorption by radiation H appears because of conservation of the energy missing from F as z increases [17]:

$$\begin{aligned} H(z) &= -\partial F/\partial z \qquad (2.6) \\ &= 2 \int_0^\infty E_{\lambda,s}\, \kappa_\lambda \int_0^{\pi/2} (1/\mu)\, f_\theta \cos\theta \sin\theta\, (1 - \rho_\theta) \exp(-\kappa_\lambda z/\mu)\, d\theta\, d\lambda \end{aligned}$$

These important expressions for F and H are developed by Viskanta and Toor [17] for the more general case including scattering and bottom reflection. Their particular application was local absorption of solar radiation in a pond. Their expressions do reduce to (2.5) and (2.6) under the assumptions made in this development.

In order to numerically evaluate the double integrals in the expressions for F

and H, the following functions of θ and λ must be determined:

(1) $E_{\lambda,\theta}$, the incident polar and spectral distribution

(2) f_θ, the fractional coverage of the hemisphere as a function of θ

(3) ρ_θ, the droplet surface reflectivity, a function of θ

(4) κ_λ, the droplet spectral absorption coefficient, a function of λ

The next four sections are devoted to this task.

2.2 Assumption of Blackbody Source

The three radiative heater coils of the experiment may be considered to be blackbodies. Then, the spectral distribution of the incident radiation is given by the familiar Planck distribution [16]:

$$E_{\lambda,s} = E_{\lambda,b} = C_1 / \{\lambda^5 [\exp (C_2 / \lambda T_R) - 1]\} \qquad (2.7)$$

where $C_1 = 2 \pi h c_0^2 = 3.742E8$ W-$\mu m^4/m^2$

$C_2 = (h c_0 / k) = 1.439E4$ μm-K

Of course, the radiative heater coils are not perfect blackbodies. However, this approximation is the only reasonable means of quantitatively characterizing the spectral dependence of the incident radiation. Furthermore, the integration step will likely smooth out any differences between reality and the blackbody assumption. The heater coil temperature T_R is read as an experimental digital display.

2.3 Analytic Calculation of Fractional Coverage f_θ

With a few reasonable assumptions analytic geometry can yield the functional form of f_θ. These assumptions are listed in the following:

(1) The axes of symmetry of the two upper cone-shaped heater coils pass

through the center of the liquid-solid interface.

(2) The elevations and radial positions of the two cone-shaped coils are equal.

(3) The ring-shaped coil is horizontal and its axis of symmetry passes through the center of the liquid-solid interface.

(4) The small eccentricity of a given point on the droplet is neglected.

Figure 3 is the vertical section of the geometry perpendicular to the faces of the cone-shaped coils with the needed measured dimensions. The measurements also implicitly assume that the two cone-shaped coils are radially opposite each other. The dimensions of Figure 4, a fabrication drawing of the cone-shaped coil assembly, are also required.

Planar trigonometry gives the f_θ function corresponding to the ring-shaped geometry in a straightforward manner as illustrated in Figure 3. Note that f_θ is constrained to be zero in a range of θ because the lip of the ring-shaped coil blocks a section of each cone-shaped coil. The analysis of the two cone-shaped coils is more involved:

(1) Find the center point of the front face of one of the cone-shaped coils.

(2) Use the radius of this disk-shaped face to find the equation of the sphere containing the circular edge of the disk.

(3) The intersection of this sphere and the sphere centered at the origin with radius equal to the distance from the origin to the point of highest elevation of the cone is the equation of the disk face of the cone-shaped coil.

(4) Write this equation of the circular disk face in spherical coordinates.

(5) Then calculate the azimuthal position of the disk ϕ_d as a function of polar angle θ. Because there are four total semicircular disk sections,

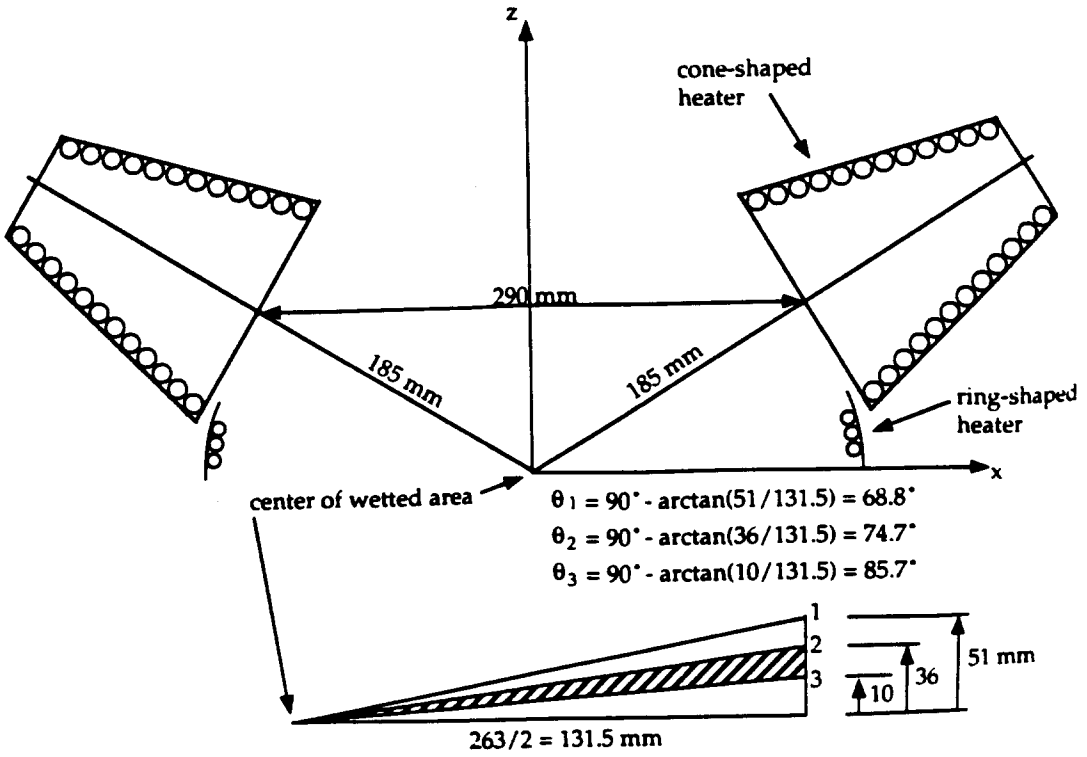

FIGURE 3
Measurements of Radiant Heater Panel Geometry

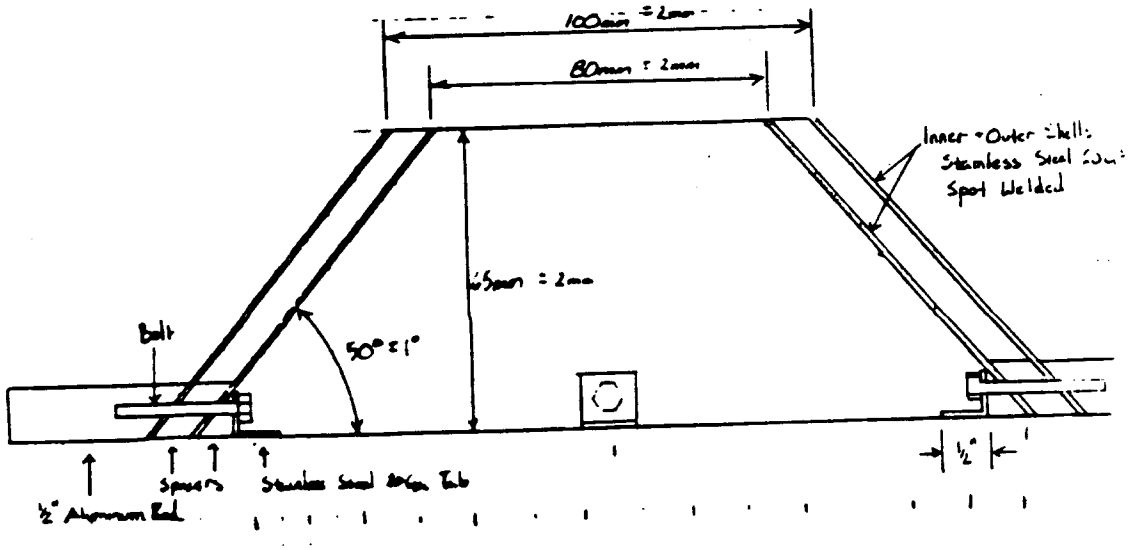

FIGURE 4
Drawing of Conical Heater Panel

$$f_\theta = 4\phi_d / 2\pi \qquad (2.8)$$

(6) Subtract out the effect of the holes in the cone-shaped coils (truncated sections) by repeating the analysis using the center of the back face of the cone and the sphere centered at the origin and passing through the edge of the back disk face.

The required nomenclature is presented in Figure 5 and the mathematical manipulations are given below.

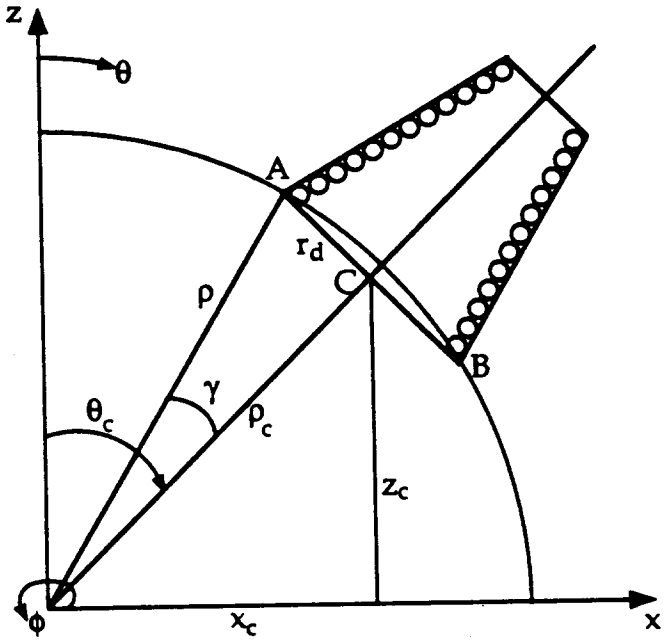

FIGURE 5
Conical Heater Panel Coordinate System and Nomenclature

(1) Choose $\phi_c = 0$

Then $C = (\rho_c, \theta_c, 0)$

$$x_c = \rho_c \sin\theta_c \cos\phi_c = \rho_c \sin\theta_c$$

$$y_c = \rho_c \sin\theta_c \sin\phi_c = 0$$

$$z_c = \rho_c \cos\theta_c$$

(2) And $(x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2 = r_d^2$

Thus the equation for the sphere centered at C is

$$x^2 - 2xx_c + y^2 + z^2 - 2zz_c = r_d^2 - x_c^2 - z_c^2 \qquad (2.9)$$

(3) The equation for the sphere centered at the origin is

$$x^2 + y^2 + z^2 = \rho^2 \qquad (2.10)$$

Then $2xx_c + 2zz_c = \rho^2 - r_d^2 + x_c^2 + z_c^2 = 2\rho_c^2$

Or $$x = -(z_c / x_c)\, z + \rho_c^2 / x_c \qquad (2.11)$$

(4) In spherical coordinates

$$\rho \sin\theta \cos\phi_d = -(z_c / x_c)\, \rho \cos\theta + \rho_c^2 / x_c$$

(5) Then $$\phi_d = \cos^{-1} [\rho_c / (\rho \sin\theta_c \sin\theta) - \cot\theta_c \cot\theta] \qquad (2.12)$$

And $$f_\theta^{front} = (2/\pi) \cos^{-1} [\rho_c / (\rho \sin\theta_c \sin\theta) - \cot\theta_c \cot\theta] \qquad (2.13)$$

where $\rho = (\rho_c^2 + r_d^2)^{1/2}$

(6) Finally $$f_\theta = f_\theta^{front} - f_\theta^{back} \qquad (2.14)$$

where $f_\theta^{back} = (2 / \pi) \cos^{-1} [\rho_c^{back} / (\rho^{back} \sin\theta_c \sin\theta) - \cot\theta_c \cot\theta]$

Of course f_θ^{front} and f_θ^{back} must be set to zero for certain ranges of θ.

Using trigonometric identities one can show that the f_θ function does in fact become zero for the uppermost and lowermost elevations of the cone-shaped coil disk face. The arccosine argument becomes equal to one:

First $\theta_{A,B} = \theta_c \pm \gamma$; $\rho_c / \rho = \cos\gamma$

Then $\rho_c / (\rho \sin\theta_c \sin\theta) - \cot\theta_c \cot\theta$

$= [\cos\gamma / \sin\theta_c - \cot\theta_c (\cos\theta_c \cos\gamma \mp \sin\theta_c \sin\gamma)] / \sin(\theta_c \pm \gamma)$

$= [\cos\gamma (1 - \cos^2\theta_c) \pm \cos\theta_c \sin\theta_c \sin\gamma] / [\sin\theta_c \sin(\theta_c \pm \gamma)]$

$= (\cos\gamma \sin^2\theta_c \pm \cos\theta_c \sin\theta_c \sin\gamma) / (\cos\gamma \sin^2\theta_c \pm \cos\theta_c \sin\theta_c \sin\gamma)$

$= 1$

Figure 6 shows the functional forms for the cones and holes. Note that the functional form is biased toward smaller polar angles; less surface area need be covered at higher elevations. Figure 7 shows the complete f_θ function after the subtraction of the holes and addition of the ring-shaped coil. The function was saved numerically in a data file for 0.1 degree increments for use in the numerical double integrations of F(z) and H(z). Note that f_θ is fairly sensitive to the geometrical measurements ρ_c and θ_c.

2.4 Reflectivity of Liquid Surface ρ_θ

Because water can be considered a dielectric, its reflectivity is strongly a function of the incident angle (already assumed to equal the polar angle θ). By assuming that the incident radiation is randomly polarized the reflectivity ρ_θ is found as the arithmetic average of the perpendicular and parallel interface reflectivities given by the Fresnel relations of electromagnetic theory [18]:

$$\rho_\theta = (R_\perp + R_\parallel) / 2 \tag{2.15}$$

Because water is weakly absorbing ($k << n$), the effect of the absorption optical constant k, and its dependence on wavelength, can be neglected. Then the Fresnel relations reduce to

$$R_\perp = [(\cos\theta - u) / (\cos\theta + u)]^2 \tag{2.16}$$

$$R_\parallel = [(n^2 \cos\theta - u) / (n^2 \cos\theta + u)]^2 \tag{2.17}$$

where $u = (n^2 - \sin^2\theta)^{1/2}$

and again $n = 1.33$ is a sufficiently accurate value representative of all

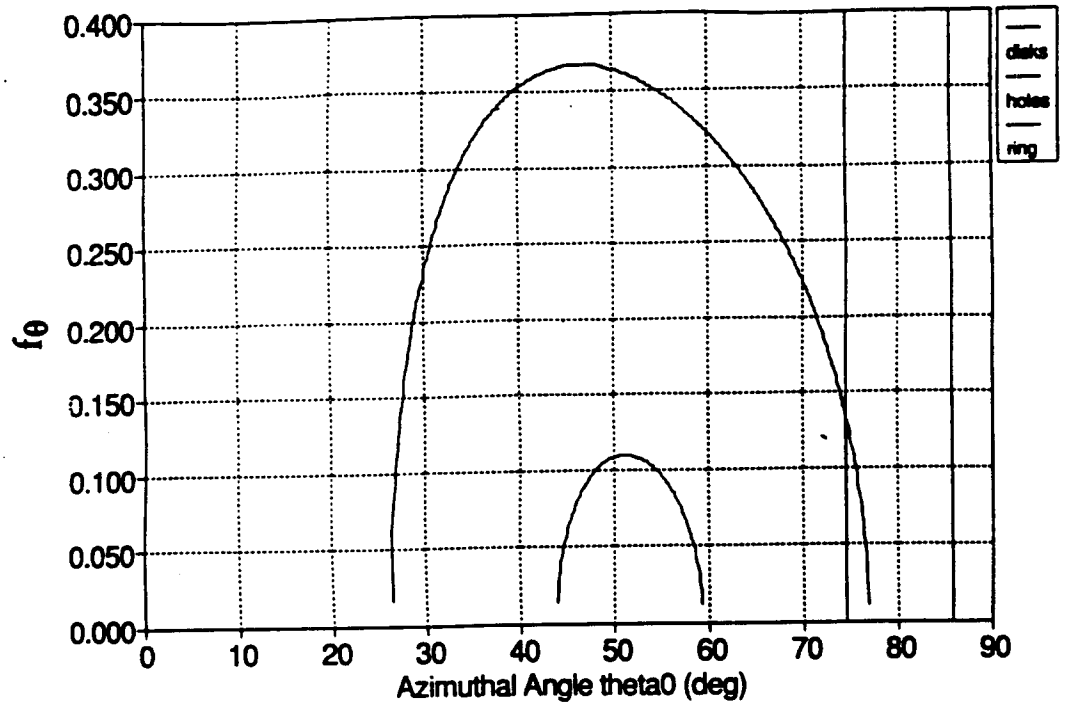

FIGURE 6
Fractional Area Coverage of Cone Face and Hole

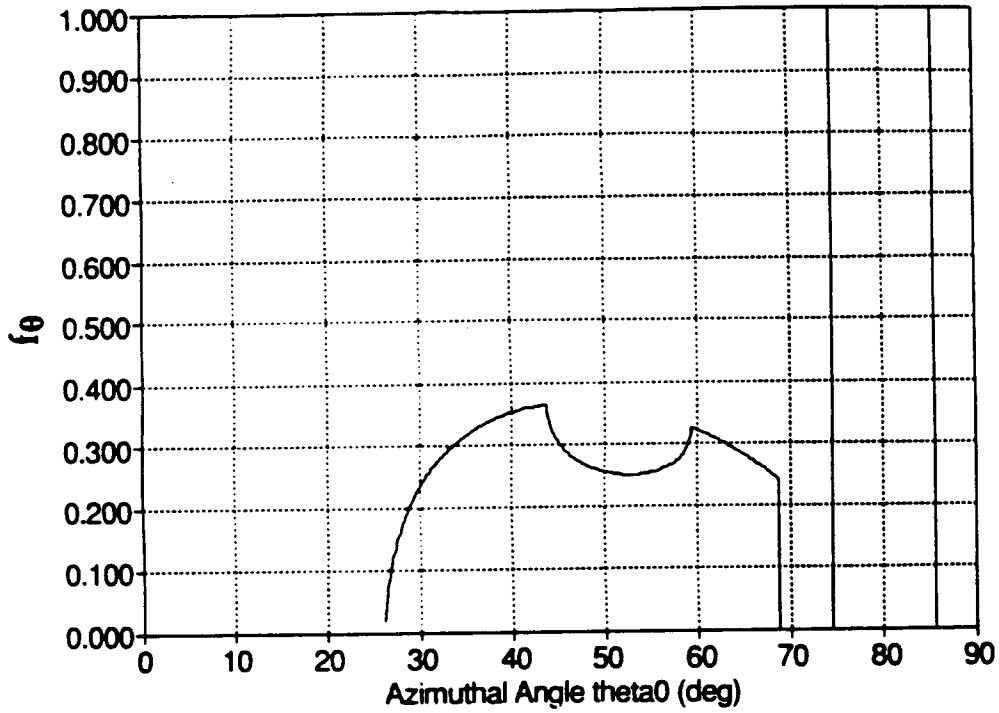

FIGURE 7
Fractional Area Coverage f_θ

wavelengths. The plot of ρ_θ in Figure 8 shows that radiation incident at shallow angles will not enter the droplet-solid system but rather will be reflected back to the surroundings.

2.5 Absorption Coefficient κ_λ

The absorption coefficient κ_λ is just the exponential damping coefficient of the radiative energy flux with distance through the absorbing medium. For water κ_λ is a very strong function of wavelength with water generally appearing opaque to large-wavelength, low-energy radiation and transparent to small-wavelength, high energy radiation. Because the blackbody distribution is also a strong function of wavelength, it is prudent to use a relatively precise form of the κ_λ function from the literature. Reference [19] gives the optical constant k for many wavelength intervals, and that data is used for the model presented here. The electromagnetic theory provides the conversion to κ_λ [18]:

$$\kappa_\lambda = 4\pi k/\lambda \tag{2.18}$$

Care was taken to accurately characterize the function for large wavelengths (> 10 μm), because a significant percentage of the energy is found in this range for the radiative heater temperatures available in the laboratory setup. Figure 9 is a log-log plot of κ_λ versus wavelength. Planck blackbody distributions are superimposed on the plot and approximate the spectral dependence of the incident energy for several values of T_R.

2.6 Numerical Integration Scheme

The numerical double integrations for F(z) and H(z) were carried out using a

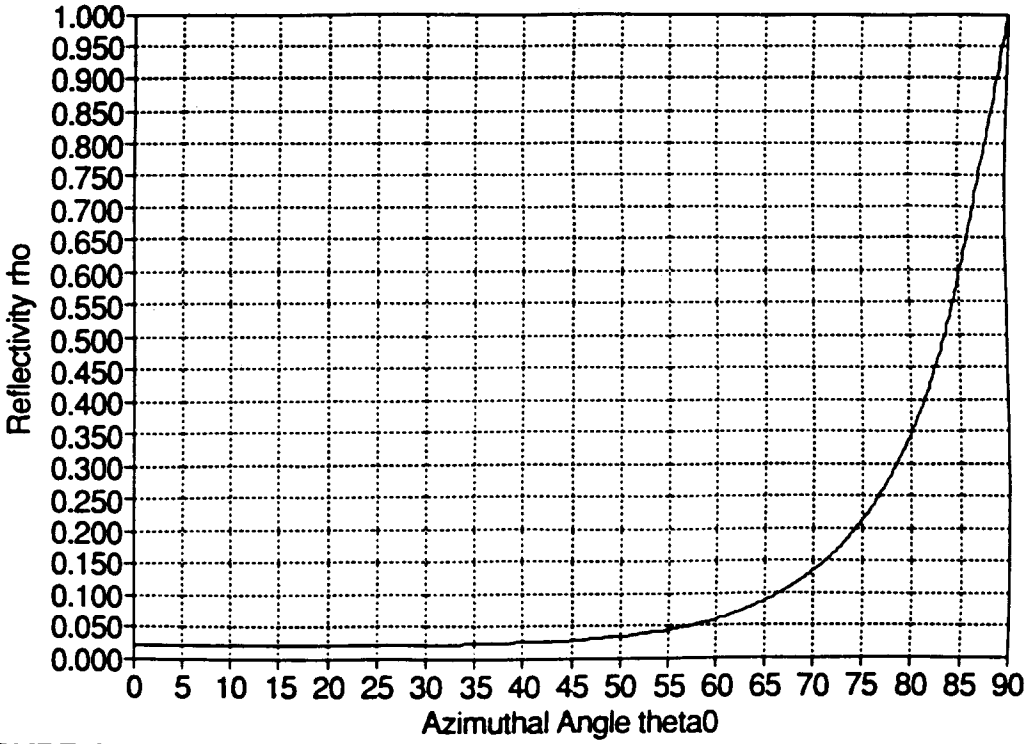

FIGURE 8
Reflectivity ρ_θ vs. Polar Angle θ

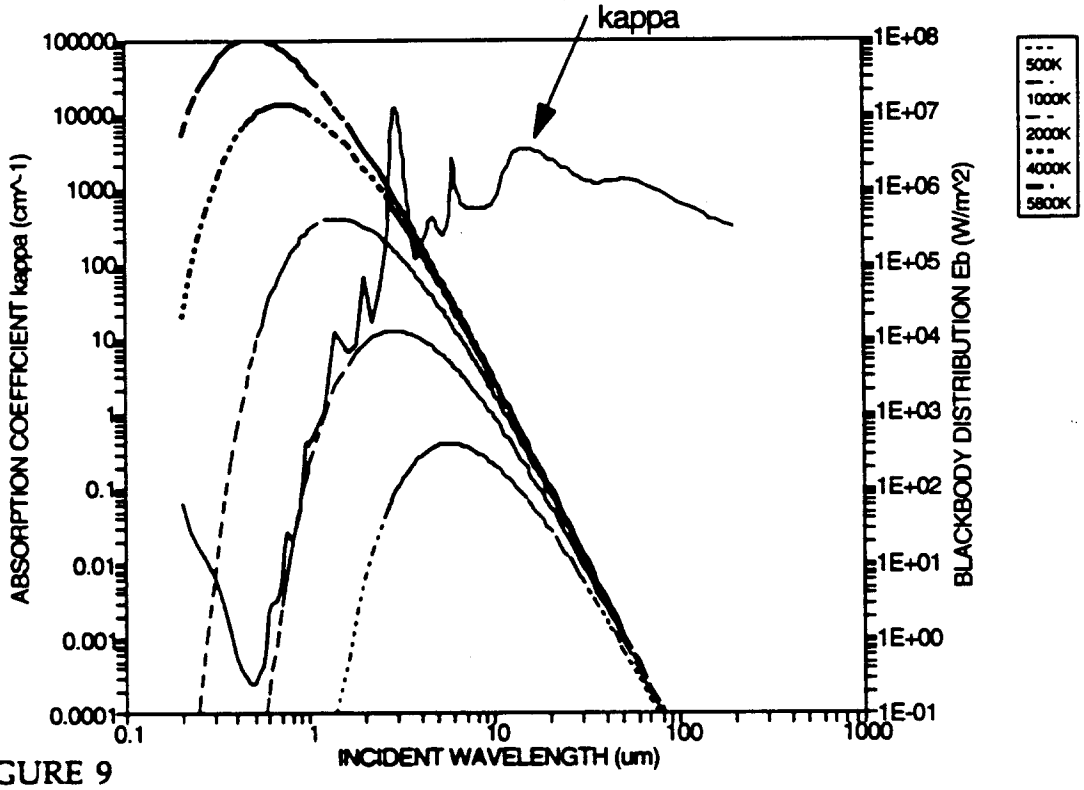

FIGURE 9
Spectral Absorption Coefficient of Water and Plank Blackbody Distributions

simple rectangular integration scheme, but the fineness of the integration grid guarantees sufficiently precise results. The step size in θ was 0.1 degree while the step size in λ varied to correspond with the optical constant data. The analytic expressions for $E_{\lambda,b}$, μ, and ρ_θ placed no additional constraints on step size. Integration over wavelength was carried out to 200 μm, where the incident radiation can certainly be neglected ($E_{\lambda,b} << 0.1$ W/m^2). The lower limit on wavelength was chosen to be 0.2 μm where again the incident radiation can be neglected, but here only for radiative source temperatures up to about 3000 K (the small bandwidth from 0.0 to 0.2 μm justifies neglecting this energy for higher temperatures). Appendix A presents a listing of the QuickBASIC computer code used in the actual computations of F(z) and H(z).

2.7 Results

Figures 10 through 12 graphically display the calculation results for the F(z) function, the radiative flux at a depth z below the liquid-vapor interface. The normalized quantities are with respect to the value at the liquid-vapor interface using the corresponding radiative heater temperature. Figures 13 through 16 display the results for the H(z) function, the volumetric heat generation term, using the same normalization scheme. Figure 16 is included only to show that H(z) truly becomes negligible for low values of T_R. The laboratory range of radiative source temperatures T_R is from 475 to 650 C (750 to 925 K) while 5800 K is the characteristic solar temperature.

Several interesting and important conclusions can be drawn from the graphs:

(1) For laboratory T_R, the radiative flux drops off to approximately 7% of its surface value at a depth of 0.1 mm.

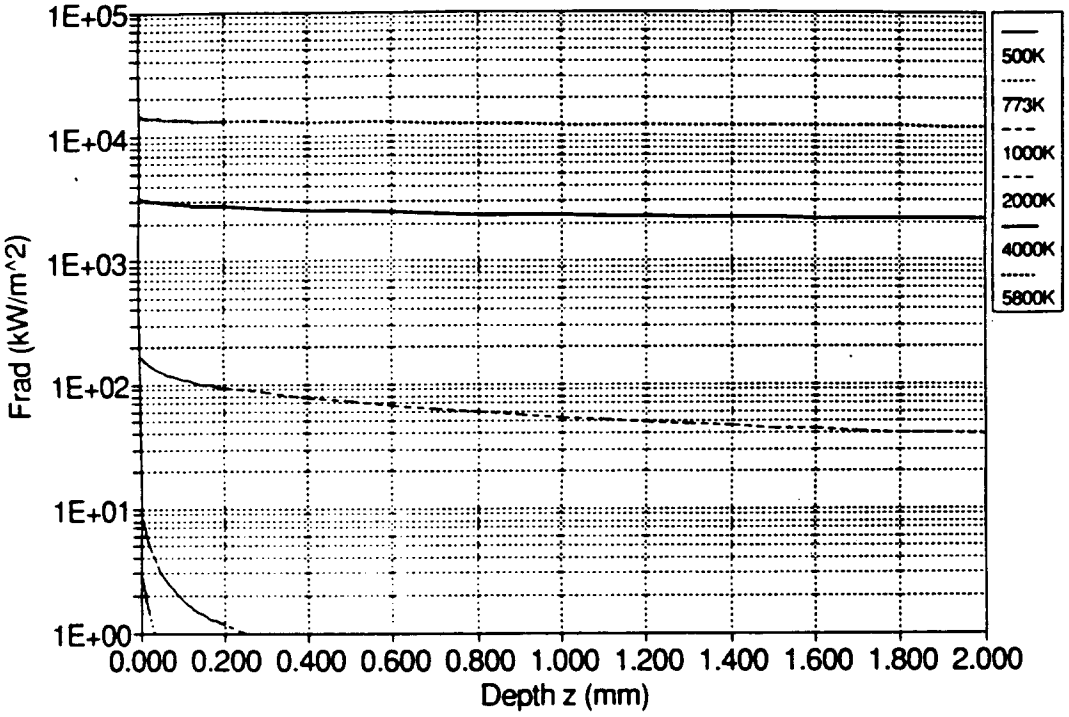

FIGURE 10
Radiative Heat Flux F(z) (z < 2 mm)

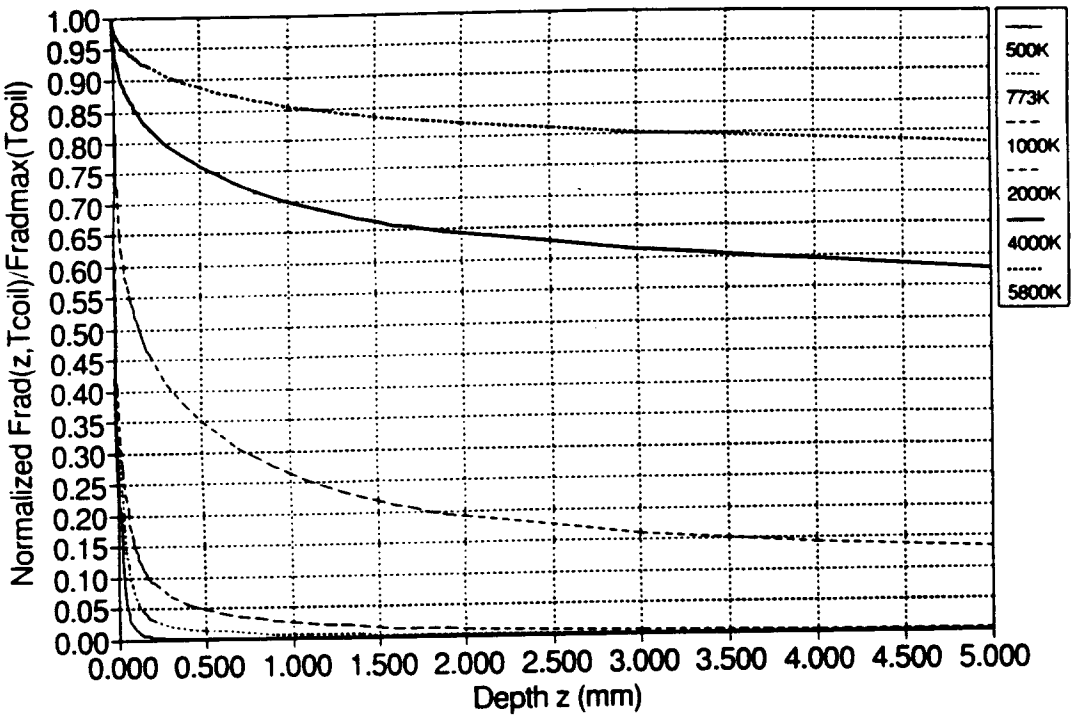

FIGURE 11
Normalized Radiative Heat Flux F(z) (z < 5 mm)

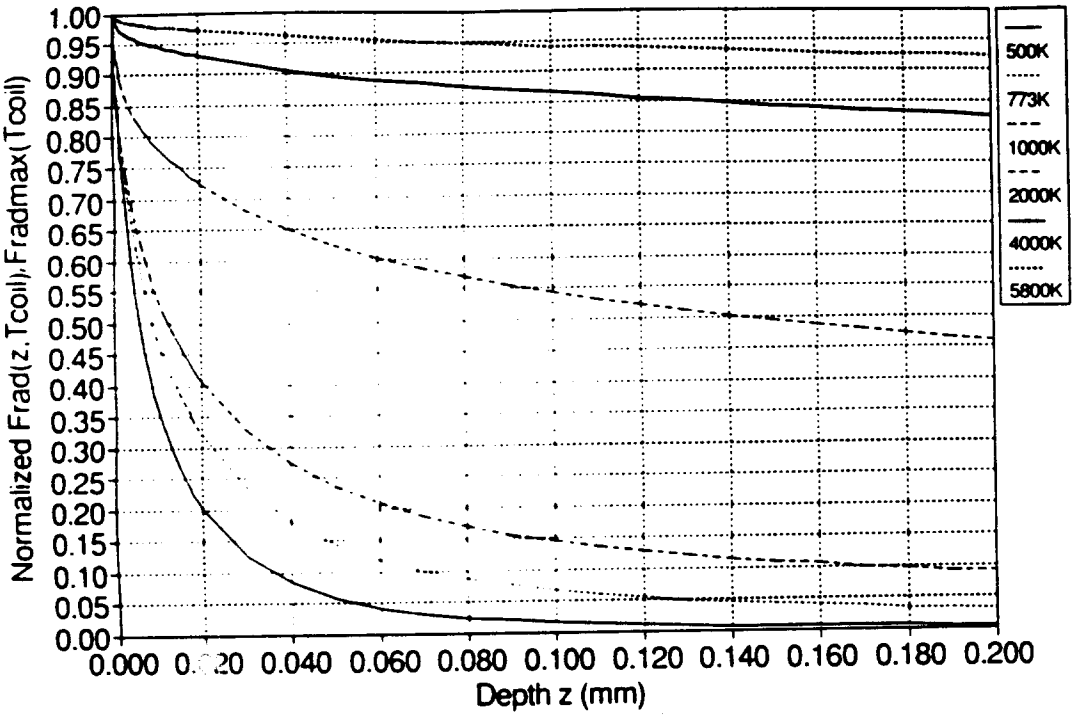

FIGURE 12
Normalized Radiative Heat Flux F(z) (z < 0.2 mm)

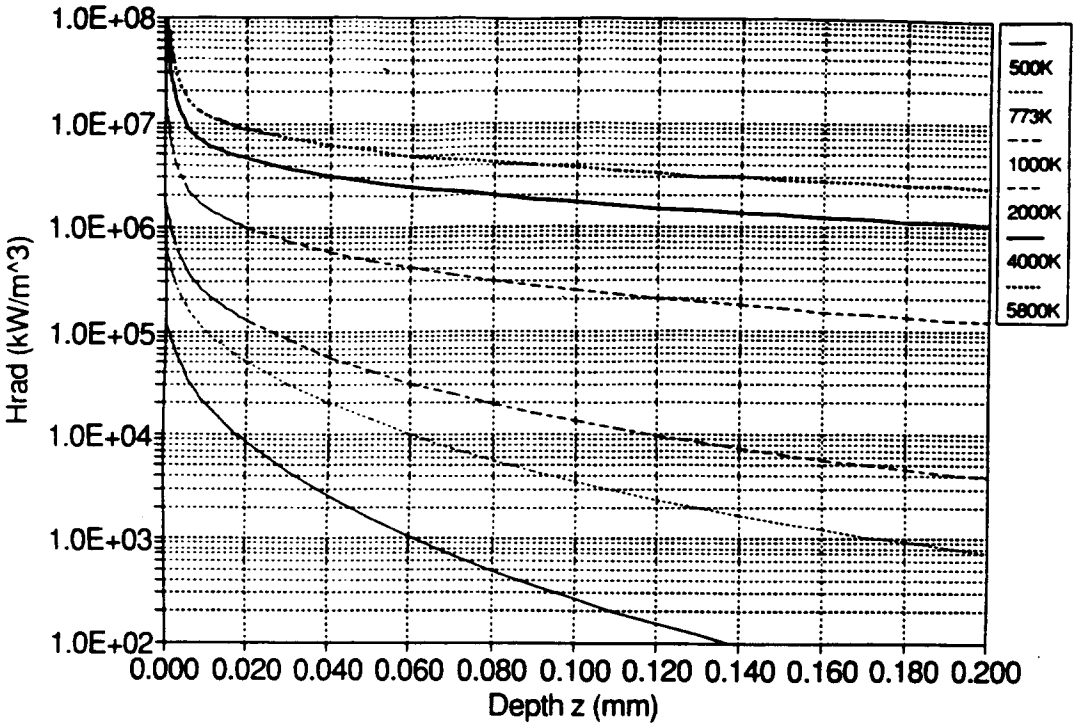

FIGURE 13
Volumetric Radiative Heat Generation H(z) (z < 0.2 mm)

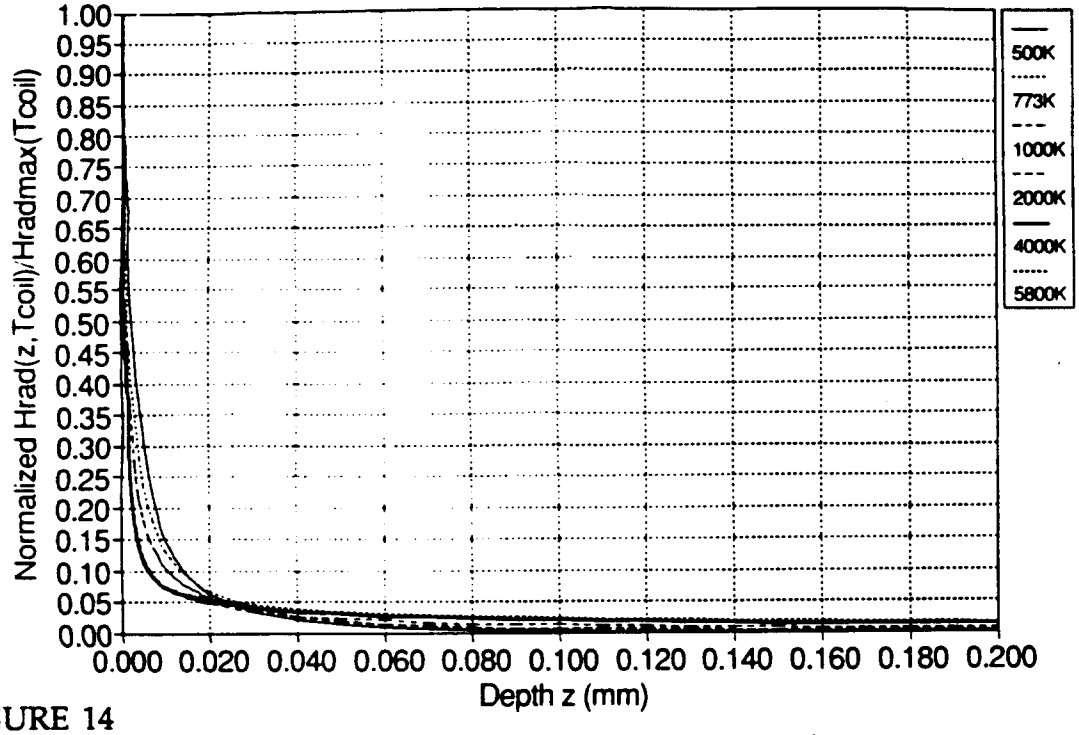

FIGURE 14
Normalized Volumetric Heat Generation H(z) (z < 0.2 mm)

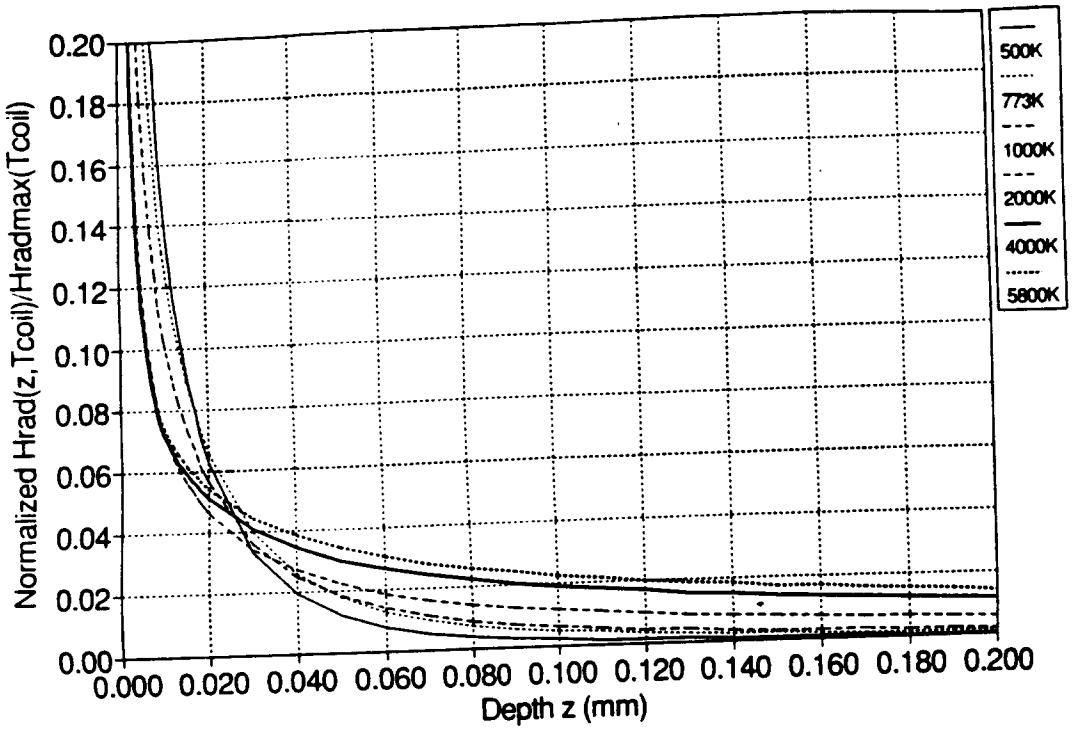

FIGURE 15
Normalized Volumetric Heat Generation H(z) (z < 0.2 mm)

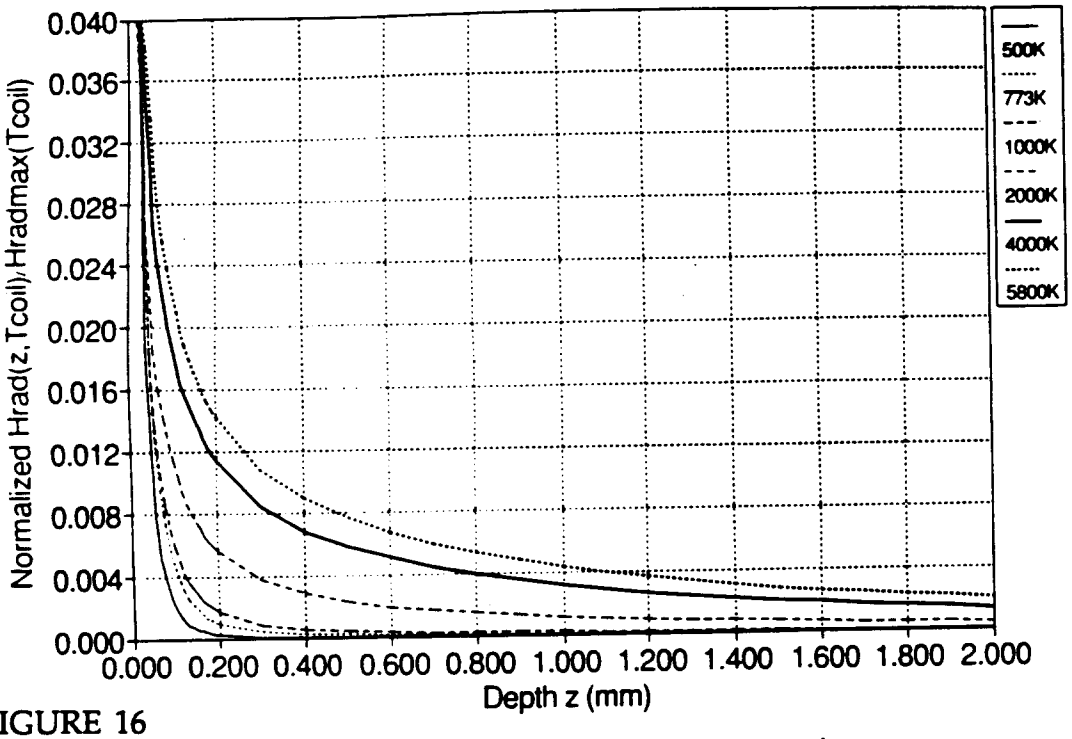

FIGURE 16
Normalized Volumetric Heat Generation H(z) (z < 2 mm)

(2) The volumetric generation term drops off to approximately 6% at a depth of 0.02 mm.

(3) For laboratory T_R, the volumetric generation term drops off to approximately 0.4% at a depth of only 0.1 mm.

(4) For higher T_R, the volumetric generation term becomes roughly linear (and to a lesser approximation constant) for z greater than about 0.2 mm.

These observations suggest that the radiative energy reaching the droplet may be input to the droplet-solid system at three locations: the liquid-vapor interface, the droplet interior, and the solid-liquid interface. In addition, for laboratory conditions, the term input to the droplet interior may be neglected. The details of the simplified model are given in the following subsection.

2.8 Simplified Model

Because the method of adding the volumetric heat generation term must be compatible with the liquid heat equation models used in the overall computer model, it is unrealistic to use the entire functional form of $H(z, T_R)$. The typical droplet has an initial apex of from one to four millimeters, so any reasonable finite difference grid could not represent the H(z) curve. The equation for the analytic quasi-steady state liquid solution, given in Chapter 3, requires two integrations of the H(z) function. These analytic integrations are not practical because of the difficult curve fit required (the next subsection only gives a piecewise curve fit). The solution is to take advantage of the very thin layer in which much of the absorption takes place and assume a surface boundary condition and perhaps a residual H term.

The radiative flux that penetrates into the droplet has a vertical value just inside the droplet given by (2.5) with z = 0. This flux is calculated with help of the quantity F_ε as defined by the author:

$$\begin{aligned} F(z=0^+) &= 2 \int_0^\infty E_{\lambda,b} \int_0^{\pi/2} f_\theta \cos\theta \sin\theta \, (1 - \rho_\theta) \, d\theta \, d\lambda \\ &= [\int_0^\infty E_{\lambda,b} \, d\lambda] \; [\int_0^{\pi/2} 2 \, f_\theta \cos\theta \sin\theta \, (1 - \rho_\theta) \, d\theta] \\ &= E_b \, F_\varepsilon \\ &= F_\varepsilon \, \sigma \, T_R^4 \end{aligned} \tag{2.19}$$

where numerical integration gives

$$F_\varepsilon = 0.2261 \tag{2.20}$$

and the Stefan-Boltzmann law has been used.

Because H(z) drops off so quickly just inside the droplet, the flux given by (2.19) can be divided into three portions: that absorbed at the liquid-vapor interface, that absorbed throughout the droplet depth, and that absorbed at the solid-liquid interface. Assuming as before no reflection at the droplet bottom, the portion absorbed at the solid-liquid interface is a fraction of the quantity given in (2.19) to be found by setting z to δ in (2.5):

$$F(z=\delta) = F_\varepsilon \, \sigma \, T_R^4 \text{ (FRACTION at } z = \delta) \tag{2.21}$$

This term is subtracted from the liquid-solution conductive flux for use by the solid BEM and will become significant when the droplet becomes extremely thin near the end of the evaporative process. The next subsection gives a curve fit for (FRACTION at $z = \delta$) so that the code can use (2.21) without storing a large numerical array of the radiative flux as a function of z

and T_R.

Figure 15 suggests that a uniform value for the residual volumetric heat generation H_c is a reasonable approximation. Then, the liquid-vapor boundary condition flux is calculated using an energy balance (per unit horizontal area):

$$F(z=0) = F_\varepsilon \sigma T_R^4 - F(z=\delta) - H_c \delta$$
$$= F_\varepsilon \sigma T_R^4 [1 - (\text{FRACTION at } z = \delta)] - H_c \delta \qquad (2.22)$$

For energy input to the droplet interior, the H_c term is also used in a nonhomogeneous term in the tridiagonal finite difference solution and in the analytic quasi-steady state solution. Note that H_c is applied in the entire droplet thickness including the thin primary absorption zone at the liquid-vapor interface. A simple curve fit for an $H_c(T_R)$ function could be carried out. Because H_c is extremely small for radiative source temperatures under laboratory conditions, the EVAP code itself assumes that H_c is zero. However, the development of Chapter 3 only assumes that H_c is uniform, so the EVAP code may easily be extended to cover higher values of T_R. Rather than assuming a constant form for the residual heat generation, a linear profile could be used. However, this would significantly complicate the quasi-steady state analytic solution, and this option is not explored.

2.9 Curve Fit for Radiative Flux $F(z, T_R)$ for Laboratory T_R

The curve fit developed here is used in the overall model FORTRAN code in order to quantify the radiation directly absorbed by the solid in the wetted region. Figure 17 suggests a hyperbolic or exponential form for a curve fit.

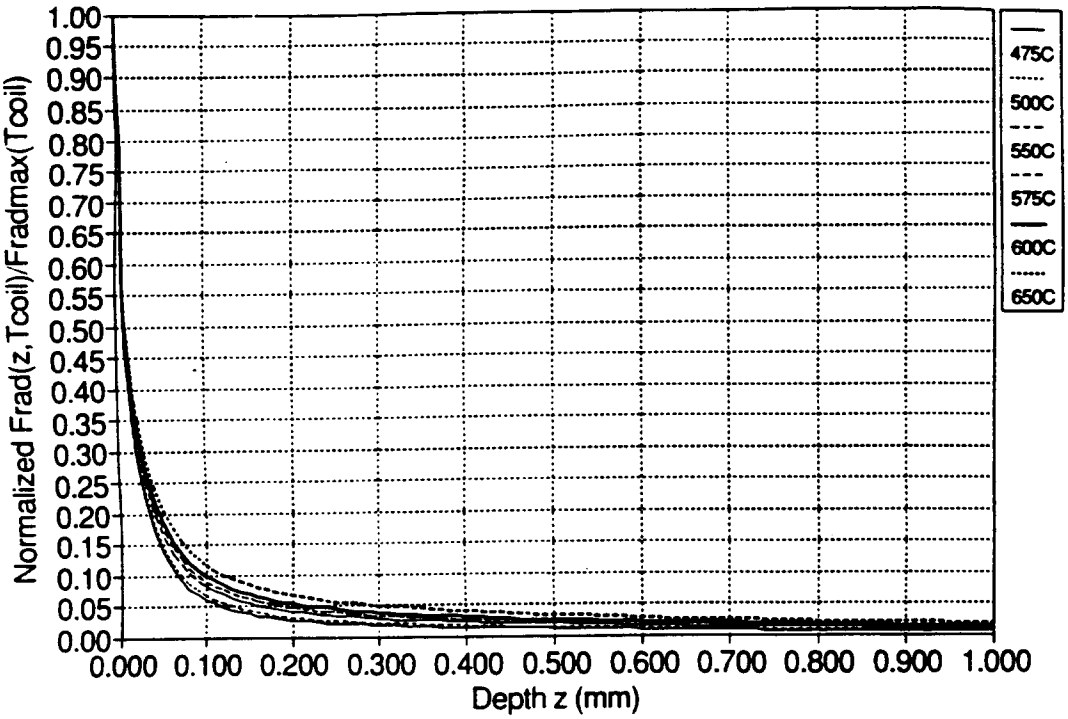

FIGURE 17
Normalized Radiative Heat Flux F(z) for Laboratory T_R

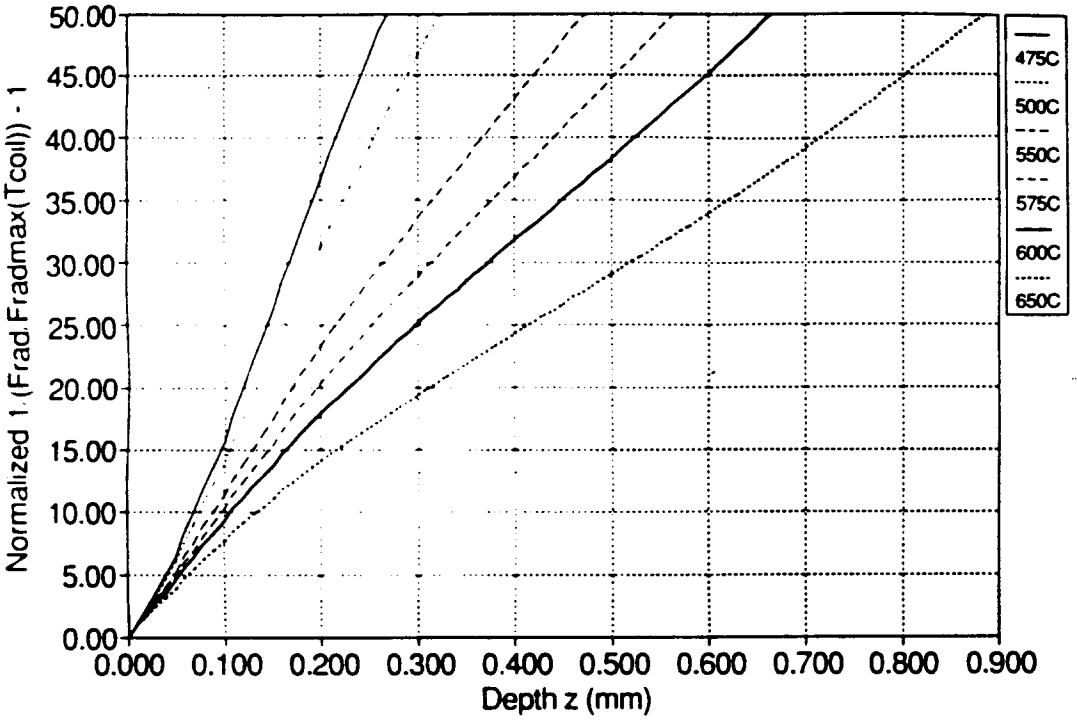

FIGURE 18
Inverse Normalized Radiative Heat Flux F(z) for Laboratory T_R

The author chooses to try a hyperbolic-type fit by plotting the reciprocal of normalized F(z) shifted down one unit (Figure 18). The six curves, representing the laboratory T_R temperatures, are indeed somewhat linear. Piecewise linear fits are made over the 1/F(z) - 1 curves over five ranges for z less than 1 mm using least-squares statistical spreadsheet functions. The range for z up to 1 mm is also used for z greater than 1 mm because the curves become rather linear at z = 1 mm and because normalized F(z) becomes rather small for z greater than 1 mm. Each linear fit for a range and T_R value results in a slope m and y-intercept b. Another spreadsheet program allowed easy least-squares second and third order polynomial fits of the m and b quantities. The resulting form of the curve fit of $F(z, T_R)$ is

$$F(z, T_R) = F_\varepsilon \sigma T_R^4 / (m(T_R)z + b(T_R) + 1) \quad (2.23)$$

$$(\text{FRACTION at } z = \delta) = 1 / (m(T_R)\delta + b(T_R) + 1) \quad (2.24)$$

where $m(T_R)$ and $b(T_R)$ are listed in the final FORTRAN code of Appendix B in the FUNCTION FRAD(D, T).

The curve fit for $H(z, T_R)$ is easily found by partial differentiation:

$$H(z, T_R) = \partial F/\partial z = F_\varepsilon \sigma T_R^4 m(T_R) / (m(T_R)z + b(T_R) + 1)^2 \quad (2.25)$$

Figure 19 shows that the original F(z) function and its curve fit evaluated for T_R of 500°C closely match with the maximum error being 3% of the liquid-vapor interfacial value. Figure 20 shows that using the curve fit in (2.25) to calculate H(z) degrades the agreement somewhat.

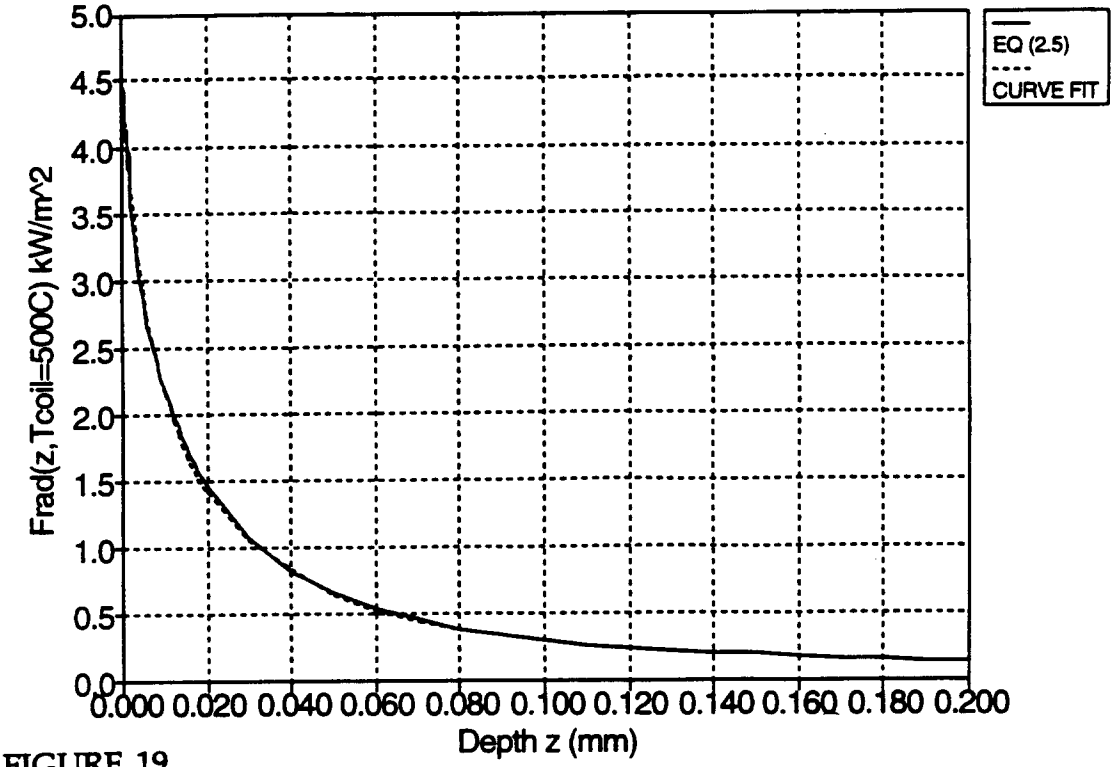

FIGURE 19
Comparison of F(z) and Curve Fit for T_R = 500°C

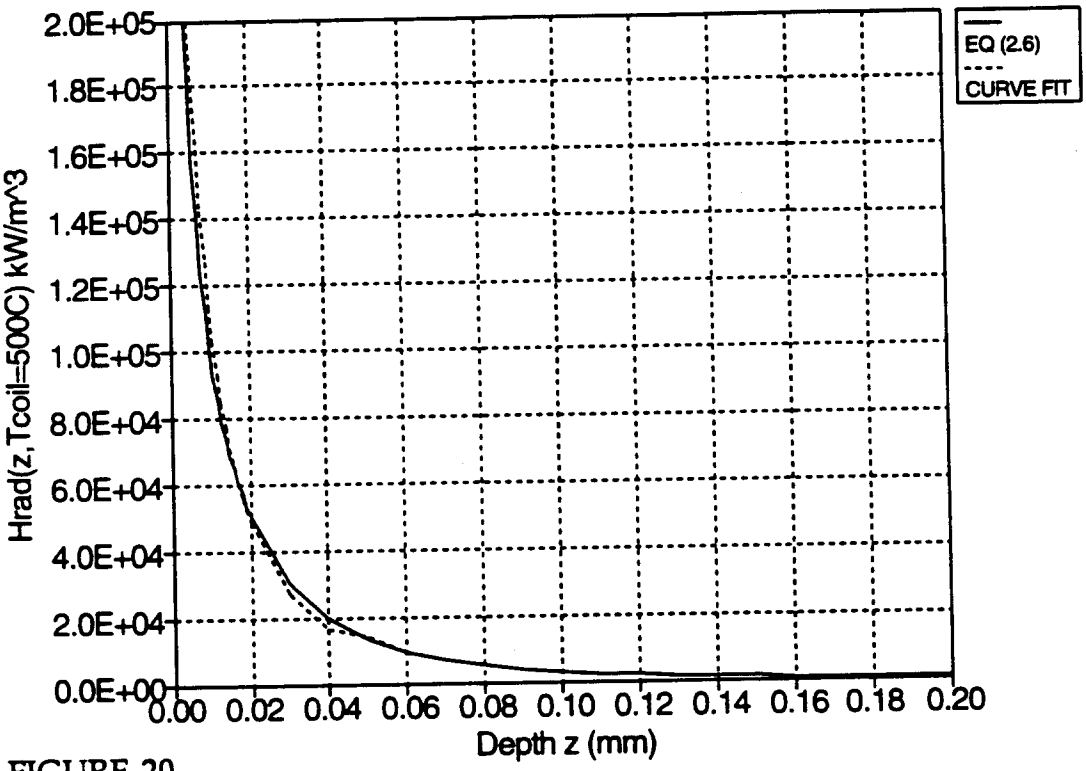

FIGURE 20
Comparison of H(z) and Curve Fit for T_R = 500°C

2.10 Summary of Assumptions and Errors

The assumptions of this chapter, both in the governing equation (2.6) and in the simplified model, have made the problem of direct radiation absorption by the droplet tenable. The assumptions and corresponding justifications used to arrive at the radiation model used in the computer calculations of Chapter 4 are summarized in the following:

(1) blackbody behavior of the radiation emitting heat panels—panels have black coatings

(2) geometrical symmetry including azimuthal symmetry of the droplet and symmetry of the radiative panels with respect to the droplet—droplet symmetry before impact and experimental design

(3) reflection at the liquid-vapor interface obeys the Fresnel relations for randomly polarized incident radiation—water can be considered a dielectric

(4) the liquid-vapor interface is horizontal and flat—droplet rather thin

(5) scattering of radiation within the droplet is negligible—droplet is relatively thin, and the water is degassed

(6) no reflection at the solid-liquid interface—emissivity of solid approximately 0.84

(7) radiation entering the droplet is completely absorbed at the surface—justified by evaluation of the very small residual volumetric heat generation for laboratory values of T_R.

Of the seven the most questionable assumption is (4) because the outward normal near the edge of the drop may in fact be closer to horizontal than vertical. Equation (2.6) then overestimates the fraction of energy reflected away from the droplet. The conclusion that under laboratory conditions

absorption is restricted to an extremely thin zone at the surface would not be affected, but two options for improvement in the overall code (Chapter 3) do become apparent:

(a) Assume some value for F_ε between that calculated with assumption (4) (i.e., $F_\varepsilon = 0.226$) and that resulting with $\rho_\theta = 0$ (i.e., $F_\varepsilon = 0.255$).

(b) Make F_ε a function of the local geometry through r and the derivative (slope) of the droplet profile.

The relatively small 13% difference between the two F_ε values suggests that using assumption (4) is not critical to the simplified radiation model so long as H_c may be neglected although this hypothesis has been put forward to explain the results of Chapter 4.

On the other hand, for real-fire non-laboratory conditions the removal of assumption (4) may affect the value of H_c and make it a function of r to be evaluated as the droplet profile is updated. These evaluations could be expected to require significant and intensive computational effort. Not only would the local reflectivity affect H, but so would the variable distances to the free surface (lengths for exponential decay).

3. SECTION II - COMPUTER CODE DEVELOPMENT

This chapter reviews in detail the development of the overall model FORTRAN code (PROGRAM EVAP) listed in Appendix B. Although the code was completely re-written, previous workers in this NIST-Center-for-Fire-Research-sponsored line of research have written codes with many similar elements [8, 10–12], and EVAP builds on their work. Re-writing allowed for maximum understanding of the effect of each level of model refinement while comparison with previous models and results provided a firm base of reference. The following program features are new:

(1) radiation model of Chapter 2 incorporating the experimental geometry of Dawson [13]

(2) framework for radiation model allowing for constant residual volumetric heat generation term—needed for high T_R

(3) exclusively one-dimensional liquid model

(4) initial liquid temperature distribution given by the error function solution to the semi-infinite solid problem

(5) avoidance of computationally intensive, extremely small initial time steps by assignment of initial quasi-steady state status to outer droplet nodes based on Fourier number criterion

(6) smooth transition from transient tridiagonal finite-difference solution to quasi-steady state analytic solution

(7) use of Clausius-Clapeyron equation in analytic linearization of evaporative liquid-vapor boundary condition

(8) option of two droplet geometry models with the second model (subroutine written by S. Tinker) allowing detailed consideration of the

transient droplet contact angle θ and the receding angle θ_r

(9) detailed reporting of meaningful transient quantities such as area-averaged upper and lower droplet surface temperatures

(10) criterion for termination of boundary element method (BEM) integration in recollection time

Furthermore, the main code, subroutines, and functions are well annotated with all variables explicitly defined, and those quantities for which modification requires additional changes in the code are clearly identified. Effort has been made to make the 1500 lines of code readable and computationally efficient. Early versions of the program were written in QuickBASIC and run on PCs with 486 microprocessors. This allowed for sophisticated debugging capabilities and computational speed. Later versions were edited using the 486s and transferred to the NIST Tiber system for execution in FORTRAN so that the required bessel and error functions could be called and the code run with acceptable speed. The final version of the code requires approximately 30 minutes to run on Tiber.

Because of high solid temperature gradients, a Green's function solution equation or BEM method is used to calculate the solid surface temperatures. The next subsection on the BEM precedes the subsection on program structure because of the strong influence BEM exerts on the structure. Subsection 3.3 describes the solid-vapor boundary conditions while subsections 3.4 through 3.9 are needed for the much more complicated solid-liquid boundary condition, i.e. the liquid solution. Subsection 3.10 shows how the BEM and liquid solutions are legitimately coupled. Finally, the last subsection details how the many property values were specified.

3.1 The Boundary Element Method (BEM)

Because EVAP is required to model solids with low thermal conductivity, the spatial and temporal temperature distributions in the solid are crucial to the evaporative process. DiMarzo [10] noted results that finite-difference methods within the solid with reasonable time steps fail because of sharp localized gradients at the droplet edge. However, a boundary element method is well suited to the problem. By linearizing the solid-vapor boundary condition, the solid problem (governing heat equation and boundary conditions) becomes linear and a Green's function approach is valid.

A heat conduction Green's function is a solution to the heat conduction problem having the same geometry but having homogeneous boundary conditions as the original heat conduction problem. It represents the temperature response at a point in time due to a heat pulse at another point at some previous time. By integrating all the heat pulses over space and recollection time, the current temperature distribution is found. Beck, et al., [20] present a very good full-text treatment of heat conduction using Green's functions in which they derive the general Green's function solution equation (GFSE) for heat conduction using mathematical manipulations such as Green's theorem. The general GFSE is

$$T(\mathbf{r}, t) = T_{init}(\mathbf{r}, t) + T_{gen}(\mathbf{r}, t) + T_{bc}(\mathbf{r}, t) \tag{3.1}$$

where T_{init} is the temperature contribution due to the initial conditions, T_{gen} is the contribution due to volumetric heat generation within the solid, and T_{bc} is the contribution due to the boundary conditions. T_{gen} is zero because

their is no internal heat generation, and T_{init} can be conveniently be made zero by subtracting away the initial temperature distribution.

The mathematical statement of the problem in the semi-infinite solid domain (see Figure 2) is

$$\partial T/\partial t = \alpha_s \nabla^2 T \tag{3.2}$$

$$t = 0: \qquad T = T_{S0} - q_0 z / k_s \tag{3.3}$$

$$0 \le r \le R;\ z = 0: \qquad -k_s \partial T/\partial z \text{ given} \tag{3.4}$$

$$r > R;\ z = 0: \qquad -k_s \partial T/\partial z = h_o (T_S - T_a) - F \varepsilon \sigma T_R^4 \tag{3.5}$$

$$z \to -\infty;\ \text{all } r: \qquad -k_s \partial T/\partial z = q_0 \tag{3.6}$$

$$r \to \infty;\ z < 0: \qquad \partial T/\partial r = 0 \text{ or equivalently } -k_s \partial T/\partial z = q_0 \tag{3.7}$$

Note that q_0 is negative for the case of radiative heat input. T_{init} is made zero by mapping the problem to the u domain:

$$u(r, z, t) = T(r, z, t) - T_{S0} + q_0 z / k_s \tag{3.8}$$

$$\partial u/\partial t = \alpha_s \nabla^2 u \tag{3.9}$$

$$t = 0: \qquad u = 0 \tag{3.10}$$

$$0 \le r \le R;\ z = 0: \qquad -k_s \partial u/\partial z \text{ given (second kind)} \tag{3.11}$$

$$r > R;\ z = 0: \qquad -k_s \partial u/\partial z = h_o (T_S - T_a) - F \varepsilon \sigma T_R^4 - q_0 \text{ (third kind)} \tag{3.12}$$

$$z \to -\infty;\ \text{all } r: \qquad \partial u/\partial z = 0 \tag{3.13}$$

$$r \to \infty;\ z \le 0: \qquad \partial u/\partial z = 0 \tag{3.14}$$

Only the boundary condition on the solid surface remains nonhomogeneous. Now using the GFSE with the expression for T_{bc} (second and third kind) yields an expression for the solid temperature distribution:

$$T(r, z, t) = u(r, z, t) + T_{S0} - q_0 z / k_s \tag{3.15}$$
$$\text{where } u(r, z, t) = \alpha_s \int_{t'=0}^{t} \int_{r'=0}^{\infty} \partial u/\partial z \, G(r, z, t; r', 0, t') \, 2\pi r' \, dr' \, dt'$$

and the proper form of the Green's function (typically found using Laplace transforms, separation of variables, etc.) is [20]

$$G(r, z, t; r', 0, t') = 2 (4\pi\alpha_s t')^{-3/2} \exp(-z^2/4\alpha_s t') \exp[-(r-r')^2/4\alpha_s t'] \, L_0(rr'/2\alpha_s t') \tag{3.16}$$

where $L_0(a) = e^{-a} I_0(a)$ is the exponentially damped modified Bessel function of the first kind of order zero and is calculated by calling a system FORTRAN subroutine.

Note that t', the recollection time, is zero at the current time step and increases in the negative time direction, i.e. $t' = t - t_0$ where t_0 is a dummy integration variable in the forward time direction. The liquid solution and infrared thermography experimental validation only require the solid surface temperatures, so the simplification z = 0 is made in the resulting expression for u:

$$u(r, z=0, t) = (4\pi\alpha_s)^{-1/2} \int_{t'=0}^{t} \int_{r'=0}^{\infty} (\partial u/\partial z) \, r' \, t'^{-3/2} \, L_0(rr'/2\alpha_s t') \exp[-(r-r')^2/4\alpha_s t'] \, dr' \, dt' \tag{3.17}$$

$$\text{where} \qquad \partial u/\partial z = \partial T/\partial z + q_0 / k_s = (q_0 - q_S) / k_s \tag{3.18}$$

The numerical evaluation of (3.17) at each time step is referred to as the BEM. Two difficulties become apparent: a) at what r do the integration terms

become negligible? and b) how to handle the degenerate cases when r or r' may be zero or equal or when t' is zero. Liao [12] showed that $r = 10R_0$ is sufficiently large for an upper bound. Zero points are avoided by a nodalization scheme in which functions are evaluated in the center of radial and time steps. The case for integration terms where r = r' is handled by an approximate analytic integration just over the element of concern in which an $\exp(u^2)$-type term becomes an erf-type term. The resulting numerical integration expression as derived by Liao [12] and verified by the author is

$$u(r, z=0, t) \tag{3.19}$$
$$= (4\pi\alpha_s)^{-1/2}\sum_{t'}\sum_{r\neq r'}(\partial u/\partial z)\, r'\, t'^{-3/2}\, L_0(rr'/2\alpha_s t')\, \exp[-(r-r')^2/4\alpha_s t']\, \Delta r'\, \Delta t'$$
$$+ \sum_{t'}(\partial u/\partial z)\, r\, t'^{-1}\, L_0(r^2/2\alpha_s t')\, \text{erf}\,[0.5\Delta r/(4\alpha_s t')^{1/2}]\, \Delta t'$$

$$\text{where } T(r, z=0, t) = T_S(r, t) = T_{S0} + u(r, z=0, t) \tag{3.20}$$

The temporal integration step size must be chosen sufficiently small to capture the temporal behaviors of the Green's function and u gradient while the overall, forward-marching time step should only capture the temporal behaviors of the surface fluxes and temperatures. The author chooses overall time steps of 0.1 s and 1.0 s with the smaller time step to be used for several seconds after droplet impact when the heat flux rapidly decreases according to the semi-infinite solution. These values are consistent with the time step chosen by Liao [12]. The products of the terms in (3.20) excluding the $\partial u/\partial z$ forcing function are considered the weight functions. Figure 21 shows the t' behavior of several typical limiting case weight functions (limiting values of r and r') where the weights which increase as t' approaches 0 apply for the case r = r'. The 0.1 and 1.0 s time steps will not adequately capture the Green's function. EVAP handles this by integrating the Green's function itself over

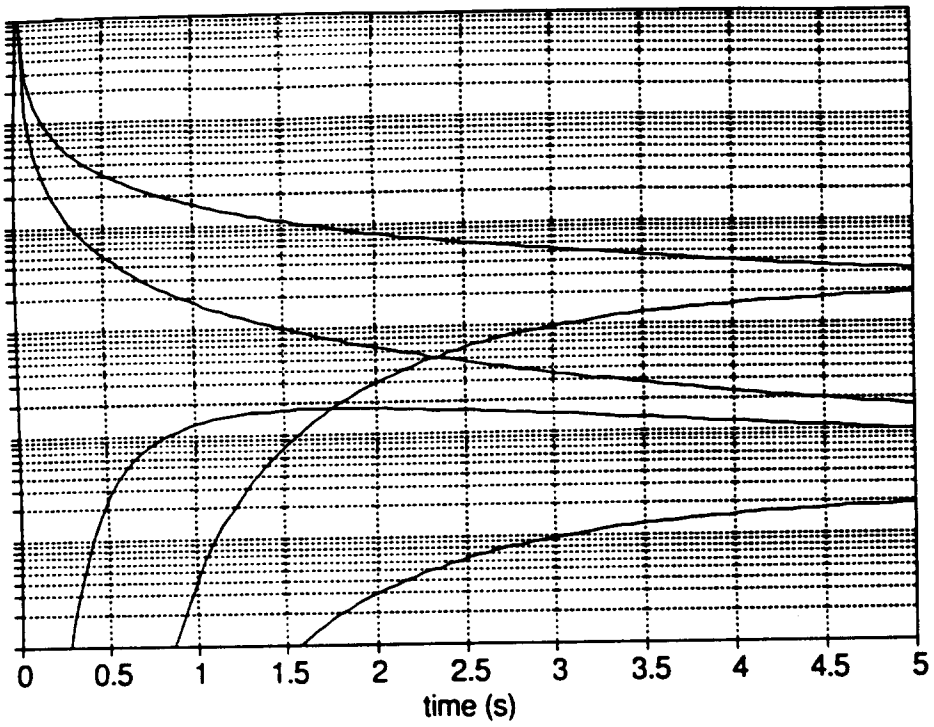

FIGURE 21
Typical Limiting-Case Weight Functions vs. Recollection Time

the first second or two using a step size of 0.004 s and saving the values in a weight matrix. For larger recollection times u is calculated at the center of the overall (0.1 and 1.0 s) time steps.

The change from the 0.1 to 1.0 s time step requires careful handling of the matrix of past forcing functions, the past solid surface heat fluxes mapped into the u domain by subtracting away the initial uniform flux. When t equals 4 s, the time step Δt is changed to 1.0 s, and SUBROUTINE RECONF is called. The algorithm in RECONF redefines the forcing function matrix FRCFNC so that the forcing function values at the centers of the 1.0 s time steps are used. The key line of code is

```
FRCFNC(K, J) = (FRCFNC(10*K - 5), J) + FRCFNC(10*K - 4), J)) / 2
```

Because the Green's function decays with recollection time, integration all the way back to droplet deposition may not be necessary. However, when the droplet becomes completely evaporated the required maximum recollection time or "memory" changes drastically. EVAP uses a dynamic measure of the required "memory" by aborting the time step integration loop if all the ratios of the current u addition to the maximum u addition (all nodes) are less than a user-defined ratio called NEGRAT. Results for the output MEMORY are given in Chapter 4.

The radial step size becomes larger for large r where temperatures and heat fluxes are known to be much more uniform. The number of elements is rather conservative at 78 with 12 making up the initial wetted area, and the outermost element extends to a distance of $13R_0$. The breakdown of element

size is given in FUNCTION WGHT.

The steps that EVAP uses to calculate the current solid surface temperature given the vector of current solid surface fluxes (u gradients) are

(1) Calculate and save integrated Green's function values for small recollection times in matrix called W.

(2) Update the matrix of past forcing functions FRCFNC using current solid surface fluxes.

(3) Call SUBROUTINE BEM1 (or BEM2) to calculate u vector as a matrix multiplication.

(4) Call FUNCTION WGHT to calculate the needed weight (effect of one element on another or itself for a certain recollection interval).

(5) For short recollection times ($\leq$1.0 s for Δt = 0.1 s and $\leq$2.0 s for Δt = 1.0 s) get the weight from W, and for longer recollection times explicitly calculate the weight using one formula for $r \neq r'$ and another for $r = r'$.

(6) Following completion of the multiplication of the weight and forcing function matrices, add u to T_{S0} to get T_S.

These functions of EVAP were validated by assuming a constant and uniform flux over a circular disk on the semi-infinite solid, a problem for which an exact solution is known. The results of this validation are presented in subsection 4.8.

3.2 Program Structure

With the requirements of the solid solution (BEM) part of the code clear, the required structure of the remainder of the code is evident. The remainder of the code must take the solid surface temperatures from the BEM and return the solid surface heat fluxes. For the non-wetted region this is a simple task,

but for the wetted area a liquid solution is needed.

The liquid solution assumes that all heat flux is one-dimensional with no radial component (previously shown to be small [15]), and proceeds through three modes: a) error function solution based on the instantaneous contact of two semi-infinite bodies; b) tridiagonal transient solution; c) quasi-steady state analytic solution. Boundary conditions consider the effects of radiative input, convection, evaporation, and re-radiation at the liquid-vapor, solid-vapor, and solid-liquid interfaces. The evaporative boundary condition requires careful calculation using the Chilton-Colburn analogy and Clausius-Clapeyron equation while the radiative boundary condition relies on the simplified model from Chapter 2. Finally, because of the critical and unpredictable effect of droplet impact, surface tension, and flow, two empirical models are used to specify the transient droplet profile. Finally, the overall model must couple the liquid and solid domains using a simple predict-correct method in which there is feedback between solid surface temperature and heat flux. Figure 22 is a simplified flow chart of the program structure.

3.3 Solid-Vapor Boundary Condition

The non-wetted solid surface boundary condition must account for radiation absorbed from the heat panels, re-radiation from the solid surface to the surroundings, and convective cooling to the ambient. The boundary condition on T(r, z, t) can then be written:

$$r > R;\ z = 0: \qquad -k_s\, \partial T/\partial z = q_{conv} + q_{re\text{-}rad} - q_{rad} \tag{3.21}$$

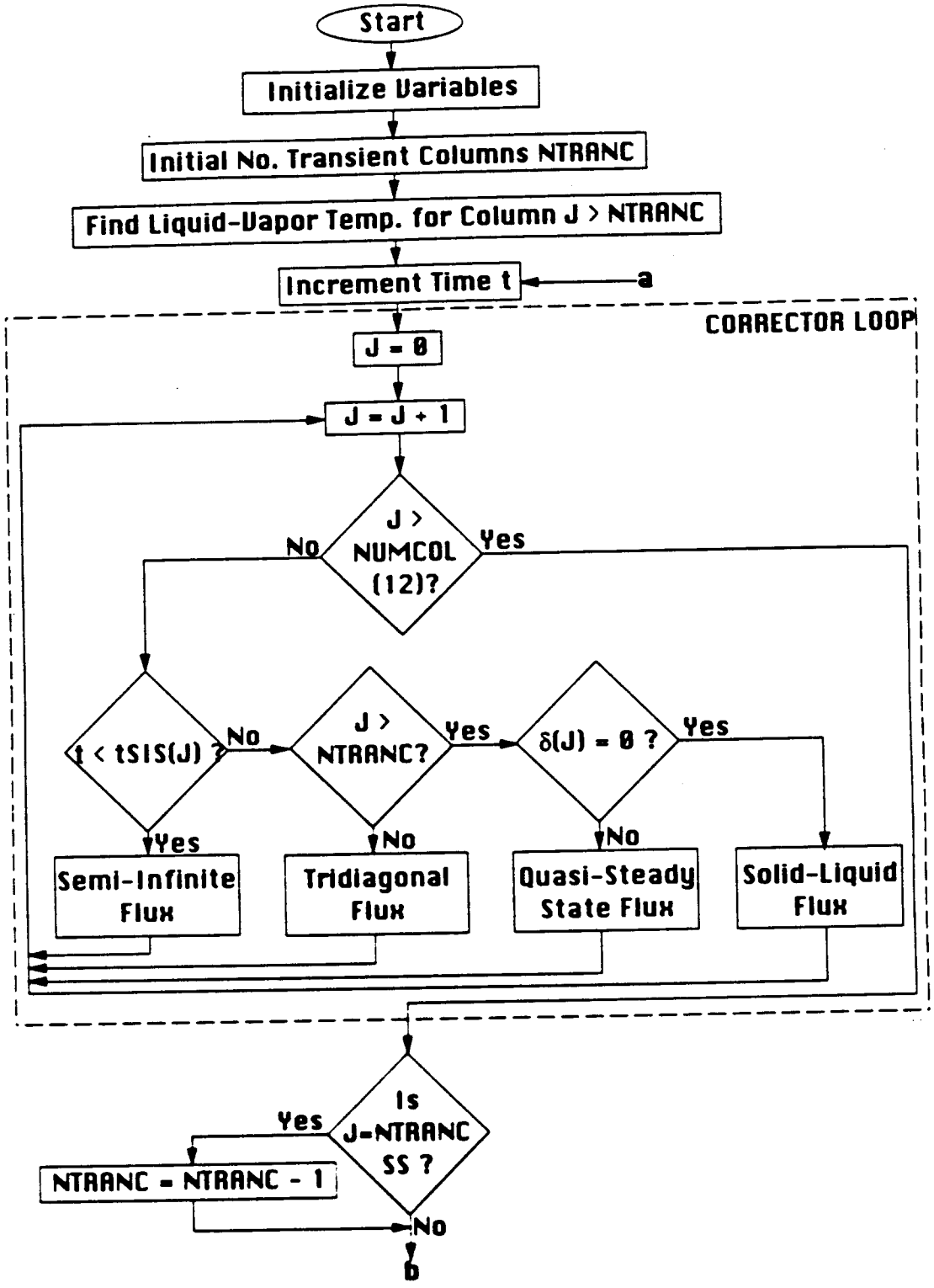

FIGURE 22
Simplified Code Flowchart

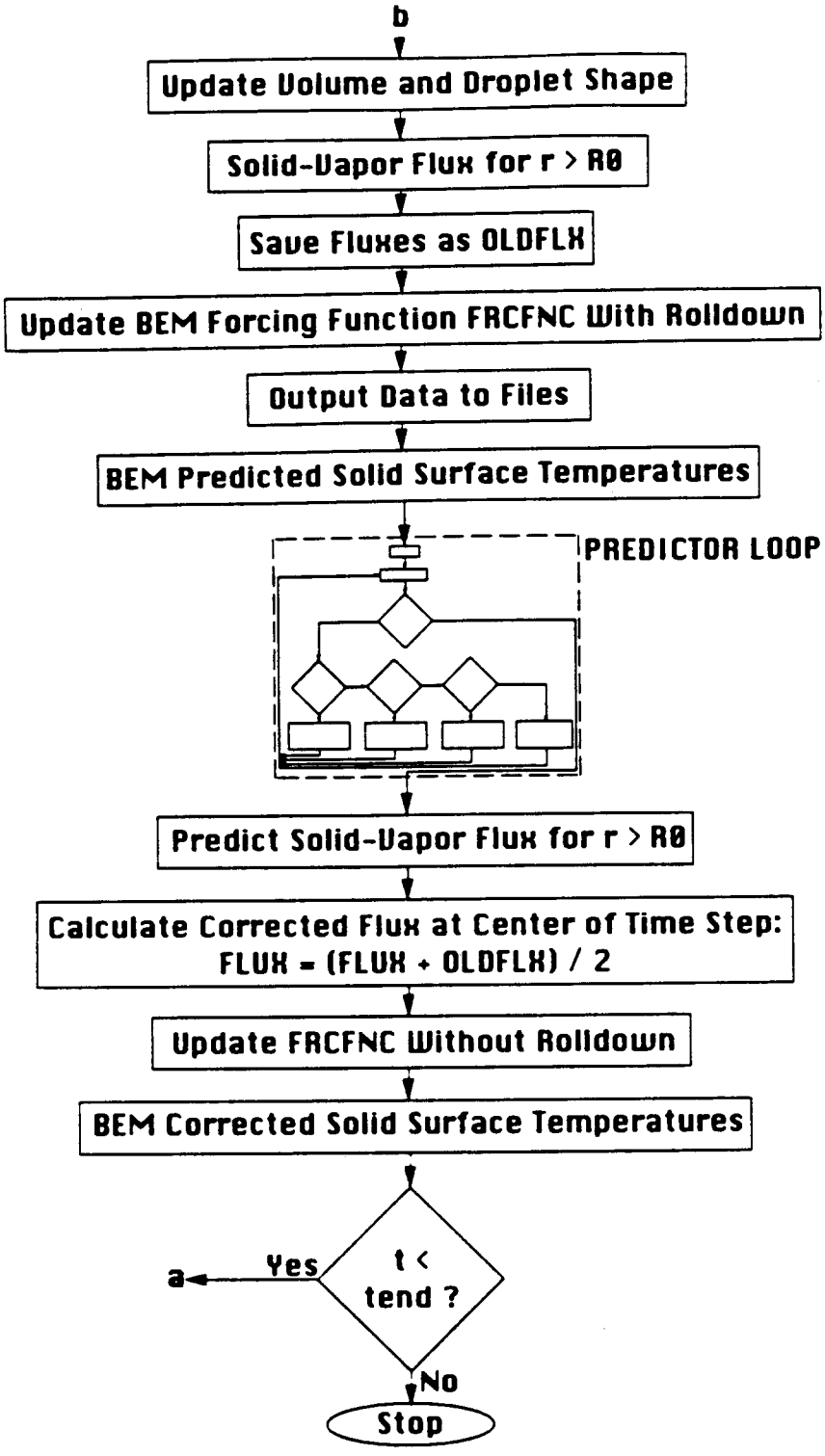

FIGURE 22 (Continued)
Simplified Code Flowchart

The convective term can be written using the experimentally measured convective heat transfer coefficient [21] as a function of initial solid surface temperature T_{S0}, curve fit by the author (see Figure 23):

$$q_{conv} = h\,(T_S - T_a) \qquad \text{where}$$

$$h = -42.348 + 1.3663T_{S0} - 0.011498T_{S0}^2 + 3.1954E\text{-}05T_{S0}^3 \ \text{W/m2-K} \qquad (3.22)$$

$$\text{with} \quad 70°C < T_{S0} < 160°C$$

The linearized form of the re-radiation term is given because the BEM requires linear boundary conditions and because the solid surface temperature is expected not to vary more than about 50°C:

$$q_{re\text{-}rad} = h_r\,(T_S - T_a) \qquad \text{where}$$

$$h_r = \varepsilon\,\sigma\,((T_{S0} - 15°C) + T_a)\,((T_{S0} - 15°C)^2 + T_a^2) \qquad (3.23)$$

and $(T_{S0} - 15°C)$ is a reasonable value of the expected solid surface temperature in order to make h_r constant. The emissivity of the Macor tile is given as $\varepsilon = 0.84$. The radiation absorption term is theoretically calculated using the f_θ term from Chapter 2:

$$\begin{aligned} q_{rad} &= 2\varepsilon \int_0^\infty E_{\lambda,b} \int_0^{\pi/2} f_\theta \cos\theta \sin\theta \, d\theta \, d\lambda \\ &= \varepsilon \left[\int_0^\infty E_{\lambda,b}\, d\lambda\right] \left[\int_0^{\pi/2} 2\, f_\theta \cos\theta \sin\theta \, d\theta\right] \\ &= \varepsilon\, E_b\, F \\ &= F\,\varepsilon\,\sigma\,T_R^4 \end{aligned} \qquad (3.24)$$

where numerical integration gives $F = 0.2552$ (3.25)

By defining the overall heat transfer coefficient as $h_o = h + h_r$, one arrives at

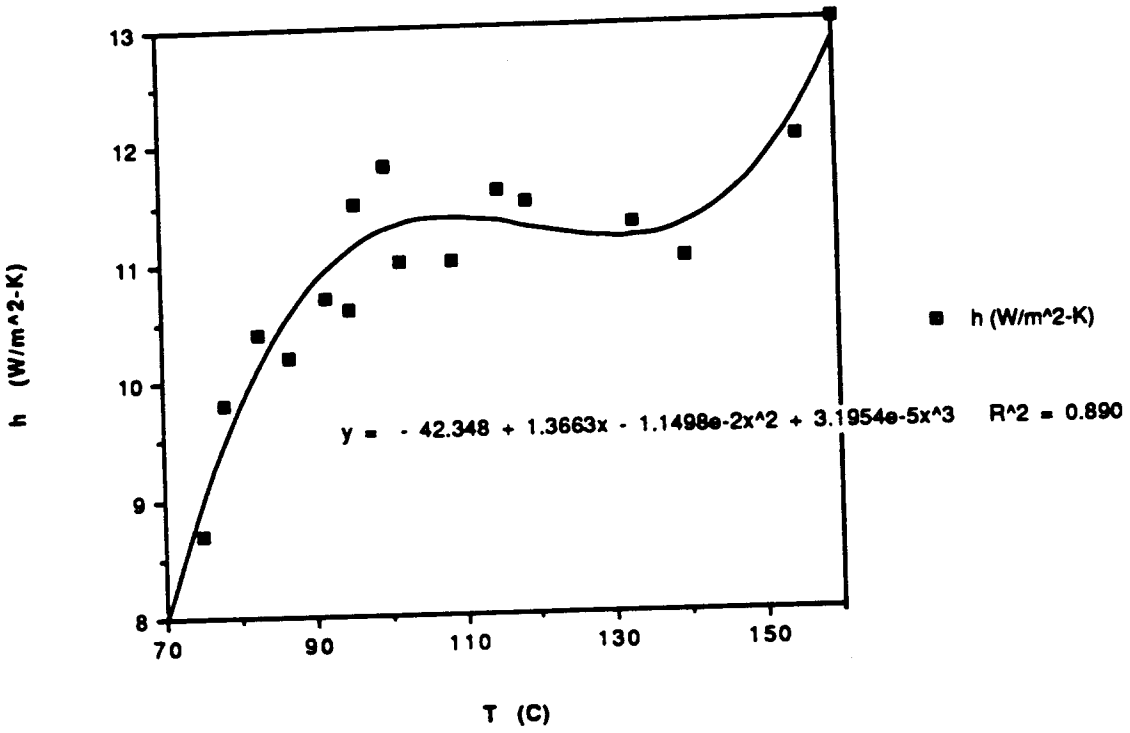

FIGURE 23
Experimental Convective Heat Transfer Coefficient vs. T_{S0} (Data from [21])

equation (3.5), the solid-vapor heat flux needed by those boundary elements not covered by liquid:

$$r > R;\ z = 0: \qquad -k_s\, \partial T/\partial z = h_o\,(T_S - T_a) - F\,\varepsilon\,\sigma\,T_R^4 \tag{3.5}$$

3.4 Semi-Infinite Liquid Solution

The purpose of the semi-infinite liquid solution is to handle the singularity of the infinite initial heat flux caused by the temperature discontinuity between the solid and just-deposited droplet. The time step required by the tridiagonal solution of the next subsection to accurately handle the high heat fluxes would be too small to be practical. However, before the diffusive heat wave reaches the liquid-vapor interface, the liquid temperature distribution will (to an acceptable approximation) be that of a semi-infinite body brought into instantaneous contact with another semi-infinite body, the solid. Another needed but reasonable assumption is that the interfacial solid-liquid solution is constant for the short time that the heat wave has not yet reached the liquid-vapor interface. Then the classic solution by Carslaw and Jaeger for a semi-infinite body initially at a uniform temperature with constant surface temperature can be used:

$$T(r, z, t) = T_C - (T_C - T_L)\ \mathrm{erf}\ (0.5z(\alpha_l\, t)^{-1/2}) \tag{3.26}$$

$$q_S = k_l\,(T_C - T_L)\,(\pi\,\alpha_l\, t)^{-1/2} \qquad \text{where} \tag{3.27}$$

$$T_C = [(k_l\rho_l c_l)^{1/2} T_L + (k_s\rho_s c_s)^{1/2} T_{S0}] \,/\, [(k_l\rho_l c_l)^{1/2} + (k_s\rho_s c_s)^{1/2}] \tag{3.28}$$

is derived by equating the two semi-infinite body surface fluxes. Note that (3.28) is strictly valid only for a constant not linear initial solid temperature. However, the heat flux from the initial linear temperature distribution is only

a small fraction of the heat flux given by (3.27), and neglecting this effect is certainly within the level of approximations already made.

This solution is enforced until the heat wave reaches the liquid-vapor interface, arbitrarily assigned to be when the error function becomes about 99.6% of its maximum value of 1.0. This percentage was chosen because it means that a 1.0 mm thick region of the droplet will require 0.4 s before the heat wave hits the upper surface. Then the time for switching to the tridiagonal solution t_{SIS} is calculated as

$$\begin{aligned} &\text{erf}\,(0.5\delta(\alpha_l\, t_{SIS})^{-1/2}) = 0.9962 \\ &0.5\delta(\alpha_l\, t_{SIS})^{-1/2} = 2.0611 \\ &t_{SIS} = \delta^2 / 17\alpha_l \end{aligned} \qquad (3.29)$$

In addition, the radiative, convective, and evaporative boundary conditions need not and cannot be applied at the liquid-vapor interface during the semi-infinite solution because conditions there have not changed since before deposition (t = 0) and are considered to be off at infinity. The time t_{SIS} varies for each wetted solid surface element because the thickness of water above each element is different. The initial liquid temperature profile for the tridiagonal solution is found using (3.26) except that T_C is replaced by the current temperature calculated by the BEM in order to keep the profile smooth as required by the tridiagonal solution for physically meaningful results. The BEM cannot function with the restriction that T_S be constant.

3.5 Tridiagonal Transient Liquid Solution

The droplet liquid can be assumed to be stagnant, and the small Eulerian

temperature time derivative associated with droplet edge recession (geometry Model B) can be neglected. Then conduction and radiation absorption are the only modes thermal energy transfer. Furthermore, the radial conduction term is small and can be neglected [15]. The resulting one-dimensional transient heat equation with volumetric heat generation and applicable boundary conditions are

$$\partial T/\partial t = \alpha\, \partial^2 T/\partial z^2 + \alpha\, H_c / k \tag{3.30}$$

$$T(z=0) = T_S(t) \tag{3.31}$$

$$\partial T(z=\delta)/\partial z = A\, T(z=\delta) + B \tag{3.32}$$

where the solid-liquid surface temperature is given by the BEM, and the liquid-vapor boundary condition has been linearized as is required to make the liquid solution linear and amenable to the tridiagonal method. Expressions for A and B will be derived in subsection 3.7.

This transient heat problem is numerically solved by dividing the droplet into annular columns with each column atop a BEM element. Each column is divided into n-2 finite difference elements with a phantom node just across each boundary. This arrangement with no nodes on the boundaries better represents the nonhomogeneous volumetric heat generation terms. Then the discrete form of (3.30–3.32) is

$$(T_i^+ - T_i) / \Delta t = \alpha\, [(T_{i-1}^+ - 2T_i^+ + T_{i+1}^+) + (T_{i-1} - 2T_i + T_{i+1})] / 2\,(\Delta z)^2 + \alpha\, H_c / k \qquad \text{for } i = 2, 3, \ldots, n-2, n-1 \tag{3.33}$$

$$(T_1^+ + T_2^+) / 2 = T_S \tag{3.34}$$

$$(T_n^+ - T_{n-1}^+) / \Delta z = A\,(T_{n-1}^+ + T_n^+) / 2 + B \tag{3.35}$$

where the second order central derivatives are averaged to take advantage of the guaranteed stability of the Crank-Nicholson method. The small temperature changes caused by the very small velocities of the nodes as they compress together as the droplet gets smaller can also be neglected. Moving the unknown new temperatures to the left in this implicit system yields

$$-\gamma T_{i-1}^{+} + (1 + 2\gamma) T_i^{+} - \gamma T_{i+1}^{+} = T_i + \gamma (T_{i-1} - 2T_i + T_{i+1}) + \eta$$

$$\text{for } i = 2, 3, \ldots, n-2, n-1 \qquad (3.36)$$

$$0.5T_1^{+} + 0.5T_2^{+} = T_S \qquad (3.37)$$

$$(-1 - 0.5A\Delta z)\, T_{n-1}^{+} + (1 - 0.5A\Delta z)\, T_n^{+} = B\, \Delta z \qquad (3.38)$$

$$\text{where} \quad \gamma = \alpha\, \Delta t \,/\, 2\, (\Delta z)^2 \quad \text{and} \quad \eta = \alpha\, H_c\, \Delta t \,/\, k \qquad (3.39)$$

An efficient standard routine, part of SUBROUTINE GAUSEL, is used by EVAP to solve this system by arranging the coefficients in an nx4 tridiagonal matrix. The residual heat generation H_c need not be taken to be unifrom in (3.39). Because H_c was found to be negligible in Chapter 2 for laboratory conditions, it is set to zero in the code.

3.6 Steady-State Liquid Solution

Although the Crank-Nicholson method is unconditionally stable, its numerical accuracy is dependent upon the value of the nondimensional multiplier $\gamma = \alpha\, \Delta t \,/\, 2\, (\Delta z)^2$. Al-Khafaaji and Tooley [22] recommend that γ be limited to 0.5. Unfortunately, as the droplet shrinks the axial step size Δz causes γ to quickly increase. However, during this time, the temperature distribution in each annular column becomes smoothed out, and the time derivative $\partial T/\partial t$ becomes negligible (as will be documented in Chapter 4). Then the liquid problem can be solved analytically yielding expressions for

the solid-liquid heat flux and updated liquid-vapor interfacial temperature. The current liquid-vapor interfacial temperature is needed to evaluate the A and B quantities in the linearized boundary condition and the evaporative mass flux. This solution is referred to as quasi-steady state because the boundary conditions and droplet thickness can still vary with time but not fast enough to warrant the need for evaluating $\partial T/\partial t$. The quasi-steady state problem statement and solution follows:

$$d^2T/dz^2 = -H_c/k \tag{3.40}$$

$$T(z=0) = T_S \tag{3.41}$$

$$dT_i/dz = AT_i + B \qquad \text{where here } T_i = T(z=\delta) \tag{3.42}$$

$$T(z) = -H_c z^2/2k + c_1 z + c_2 \qquad \text{and} \quad dT(z)/dz = -H_c z/k + c_1$$

$$c_2 = T(0) = T_S$$

$$T_i = -H_c\delta^2/2k + c_1\delta + c_2 \qquad \text{and} \quad dT_i/dz = -H_c\delta/k + c_1$$

$$\text{then,} \qquad -H_c\delta/k + c_1 = A(-H_c\delta^2/2k + c_1\delta + c_2) + B$$

solving for c_1 gives

$$c_1 = (AH_c\delta^2/2k - H_c\delta/k - AT_S - B) / (A\delta - 1)$$

$$q_S = k\,(AH_c\delta^2/2k - H_c\delta/k - AT_S - B) / (1 - A\delta) \tag{3.43}$$

$$T_i = -H_c\delta^2/2k - (AH_c\delta^2/2k - H_c\delta/k - AT_S - B)\delta/(1 - A\delta) + T_S \tag{3.44}$$

Note that H_c must be constant in the above derivation. The effect of H_c is to bow out the radiation-free linear temperature solution thus pushing thermal energy toward both interfaces. More complex forms of H_c may be used, but the resulting derivation would be somewhat complicated. Again H_c is small enough to be neglected for the laboratory conditions modelled by EVAP, so

the following expressions are substituted for (3.43) and (3.44):

$$q_S = -k\,(A\,T_S + B) \,/\, (1 - A\,\delta) \tag{3.45}$$

$$T_i = \delta\,(A\,T_S + B) \,/\, (1 - A\,\delta) + T_S = (T_S + B\,\delta) \,/\, (1 - A\,\delta) \tag{3.46}$$

3.7 Liquid-Vapor Boundary Condition and Volume Flux

Thermal energy is transferred at the liquid-vapor interface by radiation absorption (20 μm deep layer), convective cooling, and evaporation. Therefore the liquid-vapor boundary condition is

$$-k\,\partial T_i/\partial z = q_{cond} = q_{conv} - q_{rad} + q_{evap} \tag{3.47}$$

Radiation

The radiative flux assumed to be input at the surface is given by (2.22) and (2.24):

$$q_{rad} = F_\varepsilon\,\sigma\,T_R^4\,[1 - 1 \,/\, (m(T_R)\delta + b(T_R) + 1)] - H_c\,\delta \tag{3.48}$$

Convection

The droplet is assumed not to disturb the natural convection patterns established before deposition. In addition, the convective term will be seen to be only 1–2% of the total flux. Then the convective flux is

$$q_{conv} = h\,(T_i - T_a) \tag{3.49}$$

where h is given by (3.22).

Evaporation

The Chilton-Colburn analogy and Lewis number Le predict a mass transfer coefficient h_m which when coupled with the expression for total molar flux N will lead to an expression for q_{evap}:

N_l = [diffusive flux] + [bulk motion] = $J_l^* + C_l v^*$ (3.50)

using Fick's Law, $N_l = -CD\, dx_l/dz + x_l (N_l + N_{air})$ (3.51)

neglecting the solubility of air in the droplet, $N_{air} = 0$

then $N_l = -CD\, (dx_l/dz) / (1 - x_l)$

and at the interface, $N_l = -CD\, (dx_i/dz) / (1 - x_i)$ (3.52)

the definition of h_m is $J_l^* = h_m (C_{l,i} - C_{l,a})$ (3.53)

then $h_m C (x_i - x_a) = -CD\, (dx_i/dz)$

and $dx_i/dz = -h_m (x_i - x_a) / D$ (3.54)

finally $N_l = C\, h_m (x_i - x_a)/(1 - x_i)$ (3.55)

the Chilton-Colburn analogy is $h / h_m = \rho_{air}\, c_{p,air}\, Le^{2/3}$ (3.56)

and C is approximately constant (ideal gas behavior of air-vapor mixture) at the ambient value: $C \approx C_{air,a} = \rho_{air} / M_{air}$ (3.57)

also $q_{evap} = N_l \Lambda M_l$ and $M_l / M_{air} = 0.624$ (3.58)

therefore $q_{evap} = 0.624\, (h \Lambda\, Le^{-2/3} / c_{p,air})\, (x_i - x_a)/(1 - x_i)$ (3.59)

This detailed development reveals the sources of uncertainty in the evaporative term: the insolubility of air in the droplet and the Chilton-Colburn analogy and its 2/3 exponent. The property values of air, $c_{p,air}$ and Le, should be evaluated at the film temperature. The Chilton-Colburn analogy, derived by equating the functional forms of the equations relating Nusselt to Prandtl number and Sherwood to Schmidt number, makes the very

reasonable assumption that the small evaporative mass transfer rates do not influence the experimentally measured value of h. The liquid-vapor boundary condition can now be written

$$-k\,\partial T_i/\partial z = h\,(T_i - T_a) + 0.624\,(h\,\Lambda\,Le^{-2/3} / c_{p,air})\,(x_i(T_i) - x_a) / (1 - x_i(T_i)) - F_\varepsilon\,\sigma\,T_R^4\,[1 - 1 / (m(T_R)\delta + b(T_R) + 1)] + H_c\,\delta \qquad (3.60)$$

The temperature dependence of x_i in (3.60) is evaluated by making the following assumptions: a) the interface is at thermodynamic equilibrium; b) the air does not alter the partial pressure of the water vapor at the interface; and c) the air and water form an ideal gas mixture. These approximations are very reasonable for the laboratory conditions and are frequently made in psychrometrics. Then

$$x_i(T_i) = p_{sat}(T_i) / p_a \qquad (3.61)$$

where $p_{sat}(T_i)$ is given for every degree Celsius in the Keenan steam tables [23].

Linearization

Because (3.60) is nonlinear in T_i, it cannot be directly used in the tridiagonal solution. Furthermore, (3.60) cannot be used to develop an analytical expression for the quasi-steady state temperature distribution. However, the code time step is chosen to allow only small changes in the liquid-vapor temperature between iterations. Then the temperature behavior of (3.60) can be represented by a line tangent at the current value of T_i. This line is simply given by the first two terms of a Taylor series expansion:

$$q_{cond}(T_i) = q_{cond}(T_{i0}) + (\partial q_{cond}(T_{i0})/\partial T_i)\,(T_i - T_{i0}) \tag{3.62}$$

but $q_{cond}(T_i) = -k\,\partial T_i/\partial z = -k\,(AT_i + B) = -k\,A\,T_i - k\,B$

and $q_{cond}(T_i) = (\partial q_{cond}(T_{i0})/\partial T_i)\,T_i + q_{cond}(T_{i0}) - (\partial q_{cond}(T_{i0})/\partial T_i)\,T_{i0}$

therefore $A = -(\partial q_{cond}(T_{i0})/\partial T_i) / k < 0$ (3.63)

$$B = [(\partial q_{cond}(T_{i0})/\partial T_i)\,T_{i0} - q_{cond}(T_{i0})] / k \tag{3.64}$$

where $q_{cond}(T_{i0}) = h\,(T_{i0} - T_a)$

$$+ 0.624\,(h\,\Lambda\,Le^{-2/3} / c_{p,air})\,(x_i(T_{i0}) - x_a) / (1 - x_i(T_{i0}))$$

$$- F_\varepsilon\,\sigma\,T_R^4\,[1 - 1 / (m(T_R)\delta + b(T_R) + 1)] + H_c\,\delta \tag{3.65}$$

The derivative may be found numerically, but the Clausius-Clapeyron equation provides an eloquent and reliable analytic expression:

$$\partial q_{cond}(T_i)/\partial T_i = h + 0.624\,(h\,\Lambda\,Le^{-2/3} / c_{p,air})\,\partial[(x_i - x_a) / (1 - x_i)]/\partial T_i$$

but $\partial[(x_i - x_a) / (1 - x_i)]/\partial T_i = \{\partial[(x_i - x_a) / (1 - x_i)]/\partial x_i\}\,\{\partial x_i/\partial T_i\}$

$$= [(1 - x_a) / (1 - x_i)^2]\,\partial[p_{sat}(T_i) / p_a]/\partial T_i$$

$$= [(1 - x_a) / p_a(1 - x_i)^2]\,\partial p_{sat}/\partial T_i$$

$$= [(1 - x_a) / p_a(1 - x_i)^2]\,\Lambda / v_{fg}\,T_i$$

$$\partial q_{cond}(T_i)/\partial T_i = h + 0.624\,(h\,\Lambda^2\,Le^{-2/3} / p_a\,c_{p,air}\,v_{fg}\,T_i)\,(1 - x_a) / (1 - x_i)^2$$

Therefore, the needed expression is

$$\partial q_{cond}(T_{i0})/\partial T_i \tag{3.66}$$

$$= h + 0.622\,(h\,\Lambda^2\,Le^{-2/3} / p_a\,c_{p,air}\,v_{fg}\,T_{i0})\,(1 - x_a) / (1 - x_i(T_{i0}))^2$$

where, of course, T_{i0} must be in absolute units.

As shown in Figure 24, the effect of the linearized boundary condition is to

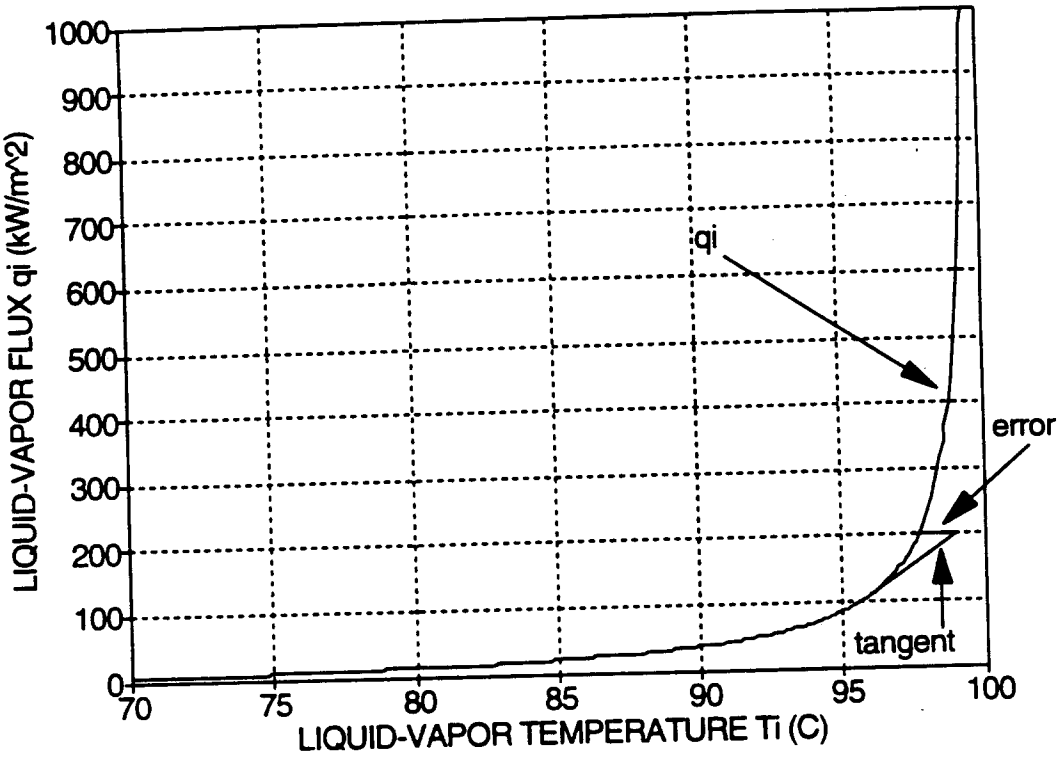

FIGURE 24
Liquid-Vapor Interfacial Flux vs. Liquid-Vapor Temperature
(assuming δ = 0.1 mm in the radiation term)

slide along the liquid-vapor flux curve and to overestimate slightly the updated T_i. For T_i greater than 95°C the curve becomes very steep and the linearization less accurate. However, the flux becomes rather stable at this point in the evaporative process, and only small temperature changes are expected. If the linearized boundary condition tries to assign an interfacial temperature greater than 100°C, then T_i is assigned a value of 99.9°C, and the liquid problem (tridiagonal transient or quasi-steady state) is solved using two boundary conditions of the first kind.

Volume Flux

Because the initial volume of the droplet V_0 cannot be measured with great accuracy, there is little benefit to considering the small volume changes of the droplet associated with temperature changes. Therefore, the droplet density is assumed to be constant at the value of water at 20°C, 998 kg/m^3. Thus the droplet inventory can be made by volume rather than mass. In addition, the small effect of temperature on the axial grid size is also neglected. The following derivation gives the volume change calculated by EVAP during each iteration in terms of the previously derived evaporative flux (3.59). The summation is made over the NUMCOL annular columns of liquid.

$$N_{l,j} = q_{evap,j} / \Lambda M_l$$

$$N_{l,j} M_l = \rho_l (\Delta V_j / \Delta t) / A_j$$

$$\Delta V_j = q_{evap,j} \Delta t A_j / \Lambda \rho_l$$

$$\text{but} \quad A_j = \pi \{ [j/NUMCOL]^2 - [(j-1)/NUMCOL]^2 \} R_0^2$$

$$= \pi R_0^2 (2j-1) / NUMCOL^2$$

$$\Delta V = (\pi R_0^2 \Delta t / \Lambda \rho_l NUMCOL^2) \sum_{j=1}^{NUMCOL} (2j-1) q_{evap,j} \qquad (3.67)$$

Again a special case must be made for the event of a false calculation of Ti greater than 100°C. The evaporative flux to be used in (3.67) is found from the known conductive, convective, and radiative fluxes:

$$q_{evap} = q_{cond} + q_{rad} - q_{conv} \tag{3.68}$$

where

tridiagonal: $q_{cond} = -k\,(T_n - T_{n-1}) / \Delta z$ (3.69)

quasi-SS: $q_{cond} = H_c\delta + k(T_S - 100°C - H_c\delta^2/2k)/\delta$ (3.70)

and $q_{conv} = h\,(100°C - T_a)$ (3.71)

and $q_{rad} = F_\varepsilon\,\sigma\,T_R^4\,[1 - 1 / (m(T_R)\delta + b(T_R) + 1)] - H_c\,\delta$ (3.48)

3.8 Liquid Model as a Function of Radial Position

The one-dimensional heat flux assumption limits the influence of one droplet on another to shape change by volume flux and to communication through the solid. The droplet shape (vector of annular droplet heights) has a strong effect on the time required until $\partial T/\partial t$ can be neglected and the quasi-steady state solution substituted for the tridiagonal solution. This required time is quantified through the nondimensional time or Fourier number $Fo = \alpha t/\delta^2$. The annular columns will become quasi-steady state starting with the thinnest, outermost column and proceeding inward. This is the rationale for the liquid solution structure as illustrated in the code flowchart (Figure 22). Provisions to change back to the transient solution are not needed because of the smoothly and slowly changing temperature profiles keep $\partial T/\partial t$ negligible. The flowchart does have the provision for when the droplet is completely evaporated (or for geometry Model B, the droplet recedes past the column) that the liquid solution is bypassed and the solid-vapor boundary condition invoked. How to decide when to invoke the quasi-

steady state solution and the scaling problem of the outermost columns are discussed below.

Inception of Quasi-Steady State Solution

For the assumption of negligible residual volumetric heat generation, the linearity of the temperature profile indicates how close the column is to the quasi-steady state solution. The ratio of solid and vapor boundary fluxes indicates the linearity. Furthermore, the solid boundary flux starts high and decreases while the vapor boundary flux starts negative (radiation overpowers evaporative flux for early, low liquid-vapor interfacial temperatures) and approaches the solid flux. Thus the mechanism for changing the solution mode is

$$FLUXR = (T_n - T_{n-1}) / (T_2 - T_1)$$

IF (FLUXR .GT. PERC/100.) THEN (PERC set by user (0.97))

SSTi = (Tn + Tn-1) / 2 (transfer liquid-vapor temperature)

NTRANC = NTRANC - 1 (decrement no. transient columns)

END IF

If H_c is not set to zero, then another approach must be used. Performing an investigative quasi-steady state solution on the outermost transient column during each time step and comparing solid boundary fluxes between the two models may be the most straightforward technique although directly calculating $\partial T/\partial t$ values is a possibility. This point must be addressed if the code is to be used with nonzero values of H_c.

Difficulty with Outermost Columns

In order to capture the transient behavior of the thinnest, outermost columns, very small and computationally expensive time steps (order of 0.01 s) must be used; the Fourier number varies with the inverse square of δ. The BEM is extremely sensitive to time step size because it requires integration back over the previous time steps for each iteration. (The number of BEM computations varies with the inverse square of time interval, and the BEM accounts for more than half of the run time.) In fact, the outermost column may reach quasi-steady sate status before the heat wave reaches the droplet apex. If too large time steps are used in the Crank-Nicholson method, large erroneous and oscillating heat fluxes will be fed to the BEM destroying the credibility of the results. Thin droplets with large β_0 are particularly a problem. Fortunately, because the transient behavior of these very thin columns is short, the BEM can tolerate flux estimates from these thin, transient columns.

These thin columns are designated as quasi-steady state in the variable initialization stage of code execution. The Fourier number is the tool used to decide how thin a droplet must be to be assigned this special treatment. As the applicable nondimensional time, the Fourier number describes how far along in the transient process is the initially uniform temperature liquid column. A related problem suggests a threshold for Fo. Consider a column of unit thickness and unit diffusivity initially at temperature zero when one boundary is forced to temperature 1. Omitting the details, the transient temperature distribution can very accurately be represented by the first term of the eigenvalue solution:

$$T(z, t) = 1 - z - (2/\pi) \exp(-\pi^2 t)$$

The time until the midpoint reaches to within 98% of its steady-state value is calculated as

$$1 - 0.5 - (2/\pi)\exp(-\pi^2 t^*) = 0.49$$
$$t^* = -\ln(0.01\pi/2) / \pi^2 = 0.421 \qquad (3.72)$$

The quantity t^* is the Fo number for this system. Experimentation with 0.421 as the threshold worked very well as there were no erroneous oscillations, and usually only the outermost one or two columns was affected. This solution to the thin-column problem works for any reasonable initial droplet geometry (β_0 and V_0).

A good initial liquid-vapor temperature is also required by the code to avoid invalid results. A poor initial value can result in the linearized boundary condition pushing the interfacial temperature past 100°C on the first time step, and recovery back down the steep flux curve (Figure 24) requires an unacceptable time. The following code is used to assign a sufficiently good initial liquid-vapor temperature:

```
10    SST_i = SST_i + 0.1
          IF (LVFLUX(SST_i) - (T_S - SST_i)/δ .GE. 0.0)  GOTO 20
      GOTO 10
20    CONTINUE
```

Again the case for nonzero H_c would add a complication here.

3.9 Droplet Geometry - Models A and B

As mentioned, the transient droplet shape must rely on an empirical model. Model A assumes a sphere segment shape and is defined by two inputs, the current volume V and the initial shape factor β_0, (or equivalently the wetted region radius R_0):

$$\beta = [\text{current radius wetted region}] \;/\; [\text{radius of sphere with equal V}]$$
$$= 2\,R \;/\; (6\,V \;/\; \pi)^{1/3} \qquad \text{(defined by Bonacina [2])} \qquad (3.73)$$

Because droplet surface tension is a function of temperature, and radiation is mostly absorbed at the droplet surface, the radiation can be expected to have a significant effect on β_0. However, EVAP takes β_0 as a measured quantity and does not directly address this issue. Model B assumes a pancake shape and also allows input of the initial solid-liquid contact angle θ_0 and a receding contact angle θ_r.

Model A

Photography of evaporating droplets suggests that the segment of sphere geometry may adequately describe the transient shape. Using analytic geometry the center of the sphere is adjusted axially until the sphere segment has the current volume V and the sphere boundary contains the solid-liquid-vapor interface. The expressions for $\delta(r)$ were developed by diMarzo and Trehan [5], checked by the author, and are repeated in the following:

$$\gamma = (a/R_0) = [4/\beta^3 + (1 + 16/\beta^6)^{1/2}]^{1/3} + [4/\beta^3 - (1 + 16/\beta^6)^{1/2}]^{1/3} \qquad (3.74)$$

$$\delta(r)/R_0 = [(1/\gamma + \gamma)^2/4 - (r/R_0)^2]^{1/2} - (1/\gamma - \gamma)/2 \qquad (3.75)$$

The dependence of the contact angle θ on the current value of β [5] is given in Figure 25. Model A updates the model shape when SUBROUTINE UPGEOA is called.

Model B

Evidence compiled by Chandra [24] shows that the droplet shape is somewhat flat, especially early in the evaporative process. Late in the evaporative process the wetted region begins to shrink as the receding contact angle θ_r is reached, and the droplet surface tension and surface adhesion forces come into balance. EVAP uses a θ_r value of 7°, a value suggested by Simon and Hsu who used experimental photographic techniques [25]. Model B has these two properties by making the following assumptions (as illustrated in Figure 26):

(1) When θ_r is reached, the droplet shape may be considered to be a segment of a sphere. This assumption is reasonable because the spherical shape minimizes surface area before shrinking of the wetted region can begin. At the inception of this shrinkage the droplet has a unique volume. The aspect ratio remains constant as the droplet recedes.

(2) The initial volume is the sum of a flat cylinder and a solid of rotation with outer surface tangent to the cylinder top face, with an acute contact angle θ_0 with the semi-infinite solid surface, and of circular arc profile.

(3) The gradual transition from initial shape to receding shape is uniquely specified by stipulating that the intersection of the horizontal tangent to the surface at the apex and the tangent to the liquid at the solid-liquid-

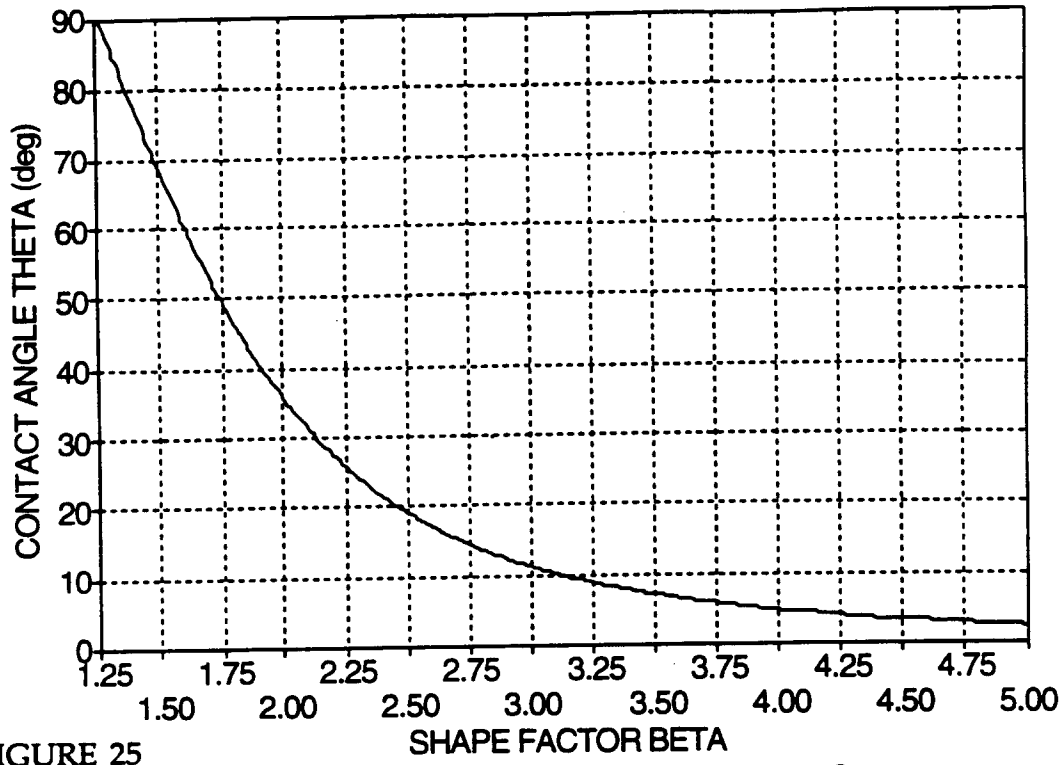

FIGURE 25
Geometry Model A Contact Angle θ vs. Geometry Factor β

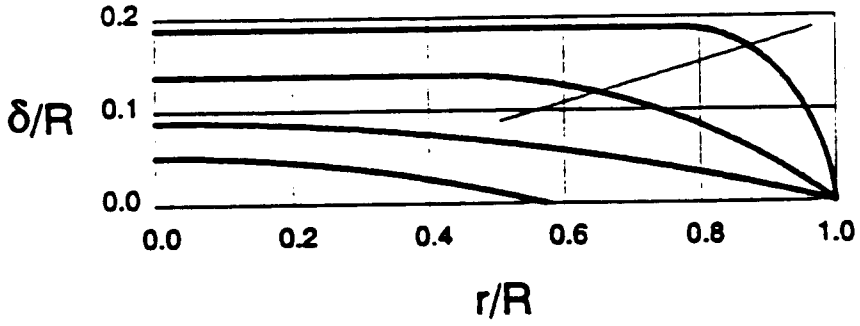

FIGURE 26
Geometry Model B Transient Configuration of the Droplet

vapor contact point is always on a straight line. This convention reasonably approximates the experimental observations.

(4) The droplet apex will always be at or less than its initial value. Surface tension cannot force the droplet to pop up against gravity during evaporation.

These assumptions have the effect of putting maximum and minimum limits on the initial contact angle θ_0. No shape meeting the four assumptions can contain the initial volume of liquid for values of θ_0 less than the θ_0 value given by the spherical segment configuration (Model A). Assumption (4) sets a maximum θ_0 for the configuration with initial apex equal to the apex at the inception of shrinkage of the wetted area. Another upper bound for θ_0 is set at 90° by Chandra [24] for evaporating and boiling droplets (nucleate boiling). A geometrical calculation shows that for β_0 less than 2.79, the maximum θ_0 as given by assumption (4) will always be greater than 90°. If one neglects the small recoil effect observed by Chandra [24], then 90° may be a good value to use. On the other hand, the θ_0 can be chosen as the value that yields the evaporation times and temperature profiles most closely matching the experimental data. Perhaps the T_{S0} and impact velocity dependence of θ_0 can be discerned this way.

Model B is invoked by calling SUBROUTINE UPGEOB, which was written by S. Tinker as part of an undergraduate project. It was slightly modified by the author to interface properly with the main routine, and the author thoroughly tested the subroutine for proper operation. The detailed workings of the 280-line subroutine are not explained here, but the code basically chooses among several potential types of droplet shape based on the

initial and current values of certain volumes.

3.10 Coupling of the Liquid and Solid Domains

The liquid and solid domains interact through the transfer of fluxes from the liquid to the solid and of solid-liquid interfacial temperatures to the liquid. The BEM solid solution uses fluxes at the center of the time intervals and gives surface temperatures at the end of the time intervals. The liquid solutions take the BEM temperatures and move up in time to the end of the time intervals producing fluxes valid at the ends of the time intervals.

The simple predict-correct method, illustrated in the code flowchart (Figure 22), solves this mismatch in times while improving the reliability of the numerical results. Rolldown of the forcing function matrix FRCFNC refers to bumping values back in time as the new value is input. Rolldown is not made in the update of the forcing function matrix following execution of the predictor loop because the next BEM call will average the predicted fluxes and last corrected fluxes (from the same time step) to get a value representative of the center of the time interval. The resulting BEM temperatures are the corrected values. Because only the integration over the first recollection interval is new to this BEM call, significant computational time (on the order of 40% of the run time without this feature) is saved by using the integration values minus the first recollection step values (vector PREDU) in the corrected BEM call (SUBROUTINE BEM2). After time is incremented, the liquid corrector loop uses the corrected temperatures and the liquid temperatures from the last corrected loop execution to calculate the corrected fluxes. (The calculated predictor loop liquid temperatures are discarded.) These fluxes, valid at the beginning of the time interval, are given

to SUBROUTINE BEM1 for calculation of predicted surface temperatures, which are then inputted to the predictor loop. The cycle continues.

3.11 Calculation of Property Values as Functions of Temperature

Rather than pick representative values for most properties, the author chose to use the available computational power to use curve fit polynomials and function calls for property value evaluation. Then the effect of using representative values can be investigated and the use of property functions discontinued if warranted. However, it is the BEM and its thousands of calls to the bessel function sublibrary that is by far the most computational intense section of code. Following are brief descriptions of the sources and curve fits of the property data. Plots and tables are given in Appendix C.

Lewis Number Le

The Lewis number, α/D, is central to the evaporative boundary condition because it allows the use of the experimentally measured convective heat transfer coefficient h to calculate the rate of mass transfer. The air value for α was constructed using data from [26] while D, given by [27], is

$$D = 0.2232\,(T/273)^n \qquad cm^2/s \qquad 273 < T < 370\ K \qquad (3.76)$$

The resulting temperature dependence for Le (Figure 48) is small, and a representative value of 0.845 at 60°C is chosen. This value compares favorably with Le number data from another source.

Constant Pressure Specific Heat of Air $c_{p,air}$

The Chilton-Colburn analogy requires this quantity. Because it varies only

0.7% over the range of interest, a curve fit is not required. However, Figure 49 gives a five-point, quadratic least-squares fit anyway. The data is from [26].

Water Vapor Interfacial Mole Fraction x_i

The temperature dependence of this quantity, p_{sat}/p_a, is crucial to the liquid-vapor boundary condition, so Table 2 was constructed for every degree C from the Keenan steam tables [23]. Figure 50 shows the exponential-like behavior of the interfacial mole fraction.

Ratio of Latent Heat to Specific Volume Change Λ/v_{fg}

This combination is from the Clausius-Clapeyron equation and is also constructed from the Keenan steam tables [23] in Table 2. Figure 51 gives the temperature behavior. Λ/v_{fg} and x_i are input to EVAP through DATA statements.

Latent Heat of Vaporization of Water Λ

The six term curve fit formula given by Bejan [28] and suggested for computer applications is used. The formula and its constants are listed in FUNCTION HFG.

Thermal Conductivity of Water k_l

Values for k every five degree C from [16] were used in the fifth order polynomial fit of Figure 52. The thermal conductivity varies 13% over the range of interest. Of course, the linear quasi-steady state model demands a single value of k, and it uses the value at the current average of the boundary temperatures. The temperature difference between droplet boundaries will be seen to be small for the quasi-steady state solution. The error function

semi-infinite solution also uses such an average value.

Thermal Diffusivity of Water α_l

The thermal diffusivity, $k/\rho c_p$, was constructed from the data in [16] using Table 3. It varies 15% over the range of interest, and the fifth order polynomial fit is given in Figure 53. Again an average value is used in the semi-infinite solution, particular to the user's choice of input temperatures.

Density of Water ρ_l

The curve fit for the density of water is not used as was discussed in subsection 3.7; the value at 20°C, 998 kg/m^3, is used. However, the temperature dependence of the density of water is included in Table 3 [16] for the calculation of thermal diffusivity. The density varies 4% over the range of interest. The fifth order polynomial fit (Figure 54) is very accurate, more convenient than the available equations of state, and may find other applications.

4. SECTION III - RESULTS AND COMPARISON WITH LAB EXPERIMENTS

After the code was fully debugged and in its final form, 20 runs were made in order to make comparisons with the single-droplet experimental results of Dawson [13] and perform a preliminary parametric study. The specific conditions for the experimental validation are

$V_0 = 10\ \mu l$; $T_{S0} = 130°C$; $T_R = 510°C$; $\beta_0 = 1.61$

for which $\tau = 24$ s on average

The β_0 value is subject to significant experimental error because of the difficulty in discerning the droplet edge in Dawson's setup: the edge was implied by the temperature line scan data. Therefore, Runs 1 through 9 are devoted to the variation of β_0. Runs 10 and 11 investigate the effect of initial volume V_0 while Runs 12 through 15 varied the initial temperatures. Use of geometry Model B was saved for Runs 16 through 19. Finally Run 20 investigated a limiting case, that of a cylindrical droplet shape with constant wetted area. The code inputs (T_{S0}, T_L, T_R, geometry model, V_0, β_0, and θ_0 (geometry Model B only)) and the principal output, the evaporation time τ, are given in Table 1. Also given are the time required for all annular columns to reach quasi-steady state status and the temporal/spatial averaged solid-liquid heat flux. The q_{disk} data refer to the constant heat flux model discussed in subsections 4.6 and 4.7. The figures in this chapter and Appendix D identify which inputs the graphed output is for by reference to run number.

Figures 27 and 28 clearly show how the droplet steadily decreases in size until it is completely vaporized. The Model A droplet maintains its spherical

TABLE 1
Results of Theoretical Calculations

RUN NO.	TS0 (C)	TL (C)	TR (C)	GEOM MODEL	V0 (ul)	BETA	R0 (mm)	THETA0 (deg)	TIME TO SS (s)	EVAP TIME (s)	FAVG (kW/m^2)	QDISK (kW/m^2)	QDISK %ERROR	PLOTS APPEND
1	130	25	510	A	10	1.61	2.152	59.838	17	49	31.201	32.3	3.5	D.1
2	130	25	510	A	10	1.7	2.272	53.212	16	44	31.093	32.3	3.7	
3	130	25	510	A	10	1.8	2.406	46.596	14	40	30.895	31.6	2.2	
4	130	25	510	A	10	1.9	2.539	40.777	13	36	30.780	31.5	2.2	
5	130	25	510	A	10	2.0	2.673	35.716	13	32	30.801	32.0	3.9	
6	130	25	510	A	10	2.1	2.807	31.346	12	29	30.835	32.0	3.9	
7	130	25	510	A	10	2.2	2.940	27.587	11	26	30.965	32.6	5.4	
8	130	25	510	A	10	2.3	3.074	24.358	10	24	31.151	32.3	3.7	
9	130	25	510	A	10	2.4	3.208	21.582	8	22	31.657	32.4	2.3	
10	130	25	510	A	9	2.0715	2.673	32.526	12	28	31.775	33.0	4.0	D.2
11	130	25	510	A	11	1.9375	2.673	38.794	13	36	29.956	31.2	4.2	
12	130	60	510	A	10	2.0	2.673	35.716	9	31	30.667	33.2	8.2	D.3
13	130	25	594	A	10	2.0	2.673	35.716	13	32	28.825	29.6	2.6	
14	120	25	510	A	10	2.0	2.673	35.716	13	41	23.445	24.1	2.7	
15	140	25	510	A	10	2.0	2.673	35.716	12	26	38.648	40.3	4.4	
16	130	25	510	B	10	2.0	2.673	60	10	38	25.166	26.3	4.5	D.4
17	130	25	510	B	10	2.4	3.208	60	2.4	30	21.275	22.6	6.4	
18	130	25	510	B	10	2.0	2.673	45	13	37	25.826	27.1	5.0	
19	130	25	510	B	10	2.0	2.673	80	7	38	24.955	26.3	5.4	
20	130	25	510	CYLIND	10	2.0	2.673	90	3.7	35	27.844	28.9	3.8	

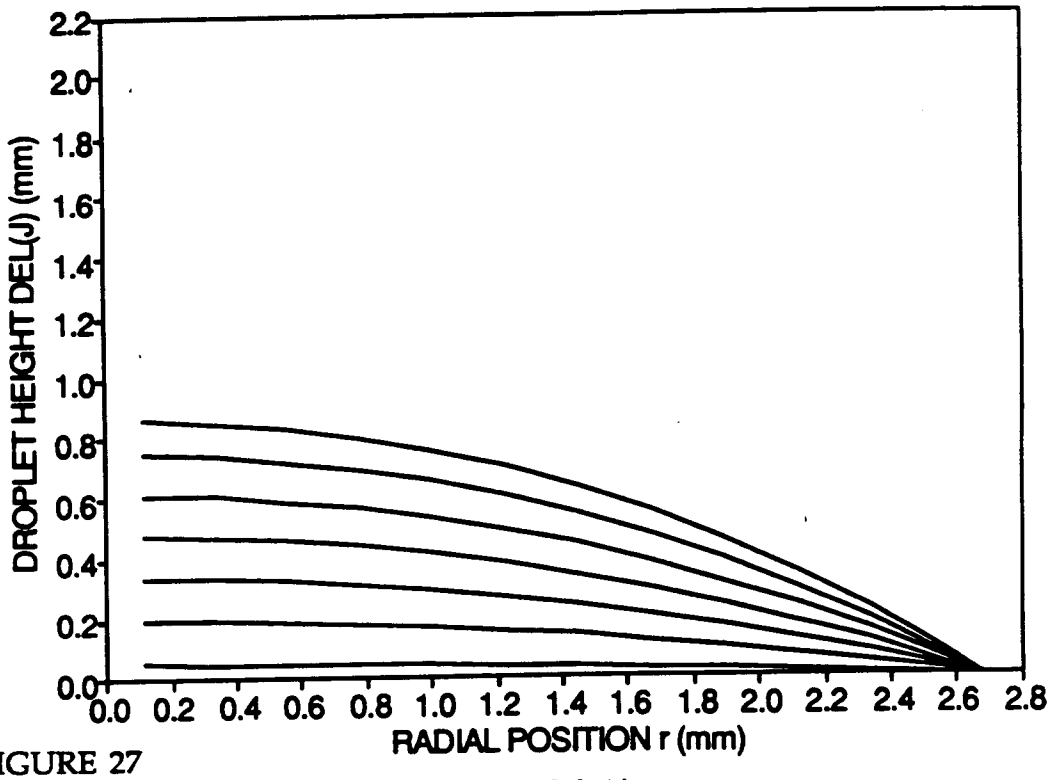

FIGURE 27
Droplet Height: Run 5 (Geometry Model A)

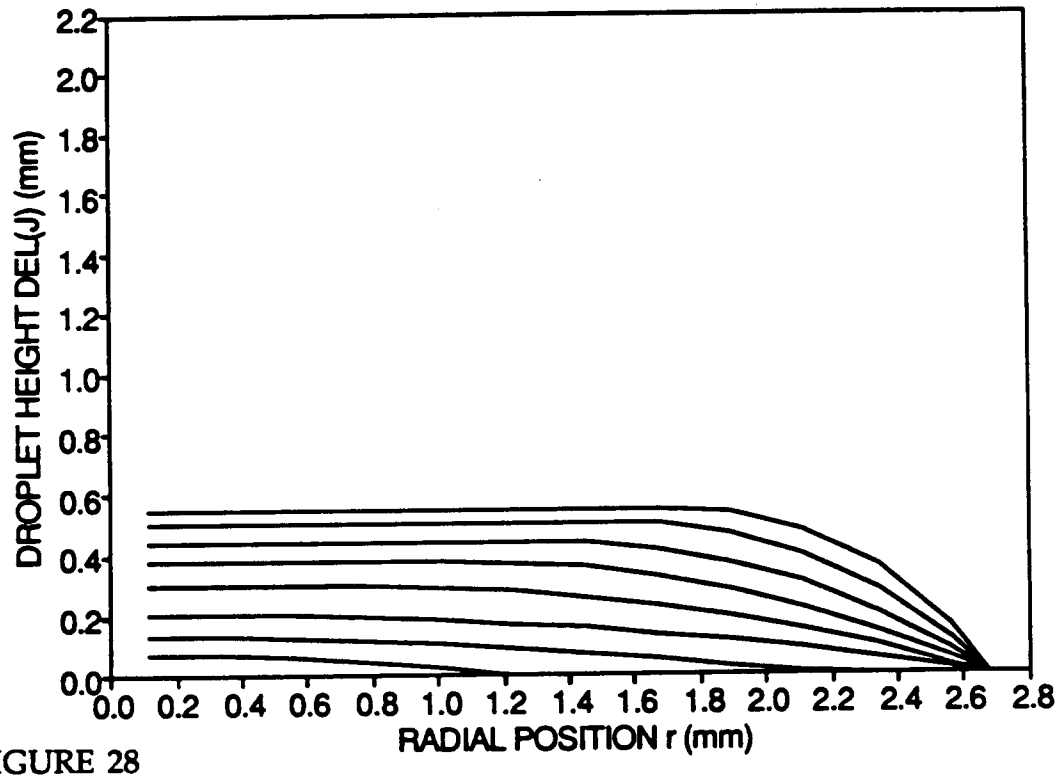

FIGURE 28
Droplet Height: Run 16 (Geometry Model B)

segment shape at all times while the Model B droplet displays the more complex behavior for which it was designed.

Figure 29 shows the cumulative energy transfers across the droplet boundary that must in the end add up to the latent plus sensible energy change in the original droplet mass. The conduction component is initially high because of the large temperature gradient at the solid interface. The direct radiation component is a almost straight line because it is not a function of any droplet temperature (it is a slight function of droplet thickness). The percentage of energy transfer by radiation is mostly just a function of wetted area and evaporation time since the flux is nearly constant. The percentage is approximately 15% for the run illustrated.

Figures 30 through 40 display the temporal and spatial behavior of the solid-liquid flux, BEM forcing function, liquid-vapor temperature, and solid surface temperature for one case each of geometry Models A and B. The heat fluxes are highest at the droplet edge simply because the droplet is thinnest there. The effect is more pronounced for Model A as expected because of the larger very thin region. The forcing function graphs are just the flux graphs inverted and translated, but they do show how small the forcing function becomes outside the wetted area. There the only contribution is the convection and re-radiation due to the temperature depression from initial conditions. The oscillating behavior of the flux and forcing function for early times near the droplet edge originates from the discreteness of the numerical grid and would diminish with decreases in the radial and time steps.

The liquid-vapor temperature graphs show that the upper surface is hotter at

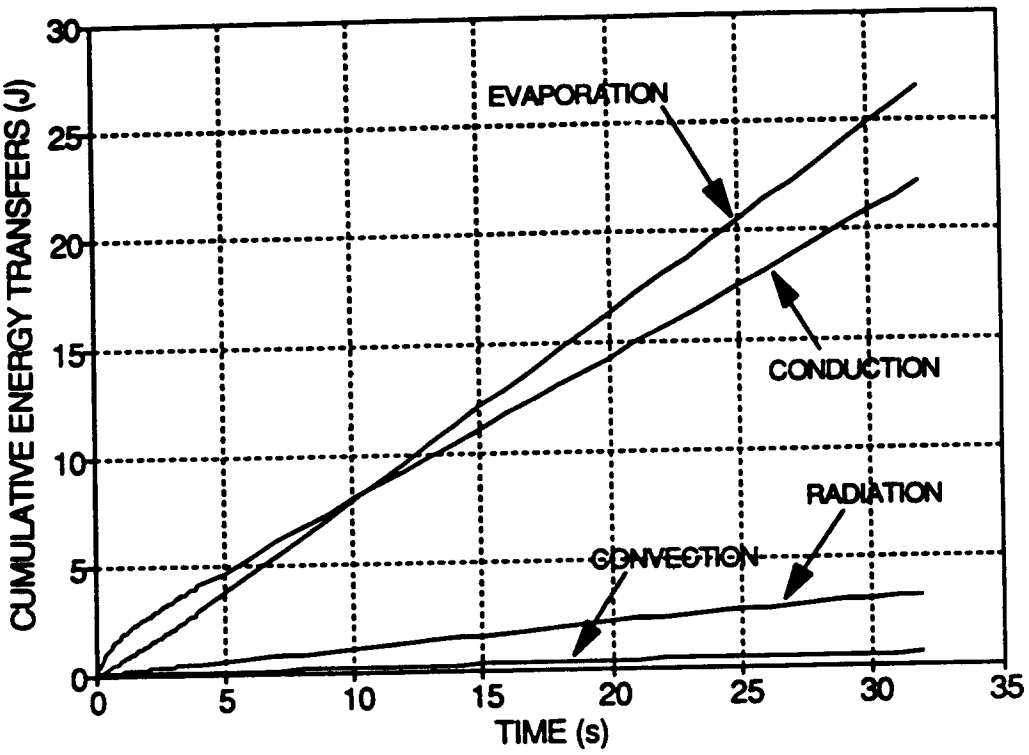

FIGURE 29
Cumulative Energy Transfers: Run 5

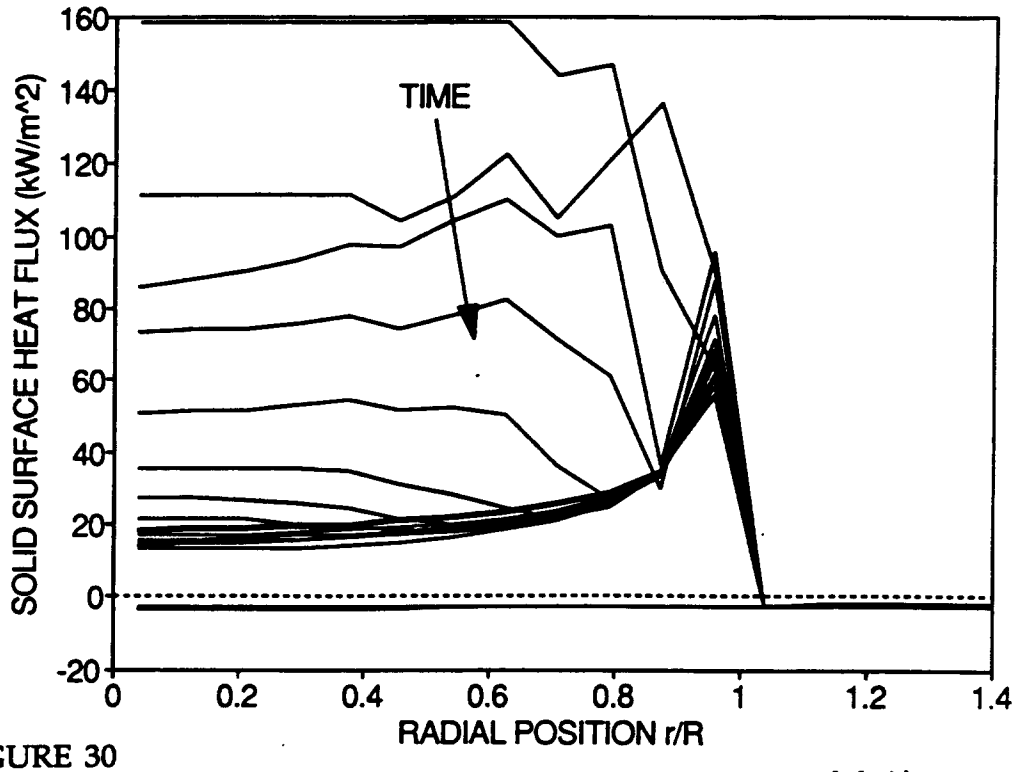

FIGURE 30
Spatial Profile of Solid Surface Flux: Run 5 (Geometry Model A)

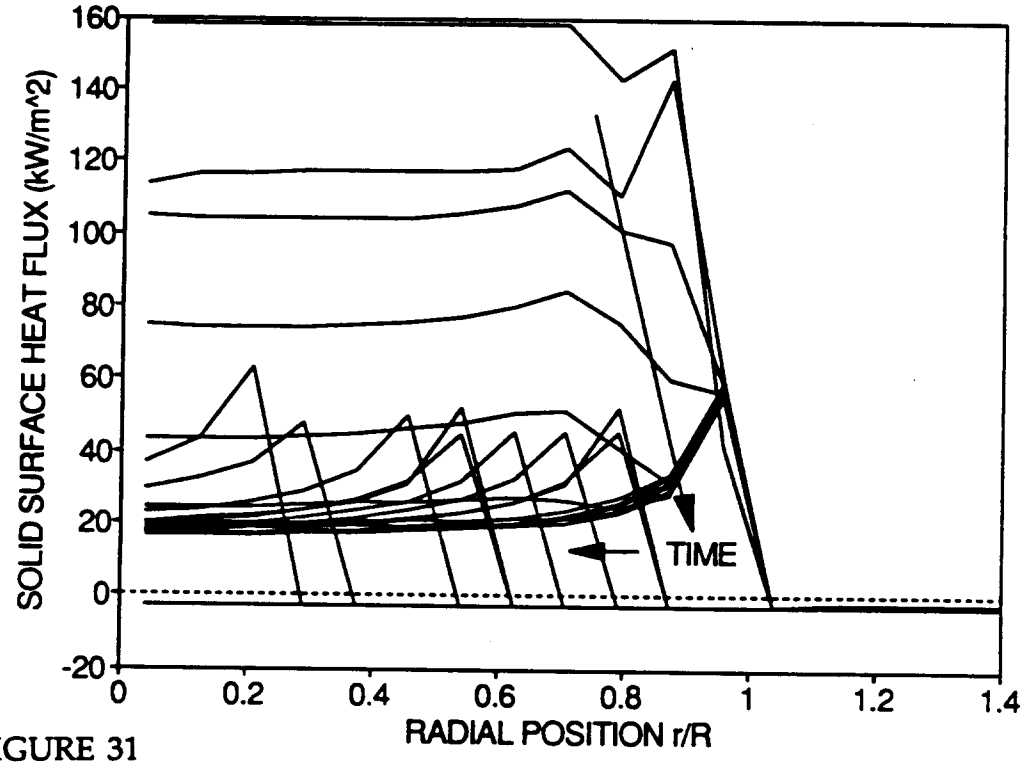

FIGURE 31
Spatial Profile of Solid Surface Flux: Run 16 (Geometry Model B)

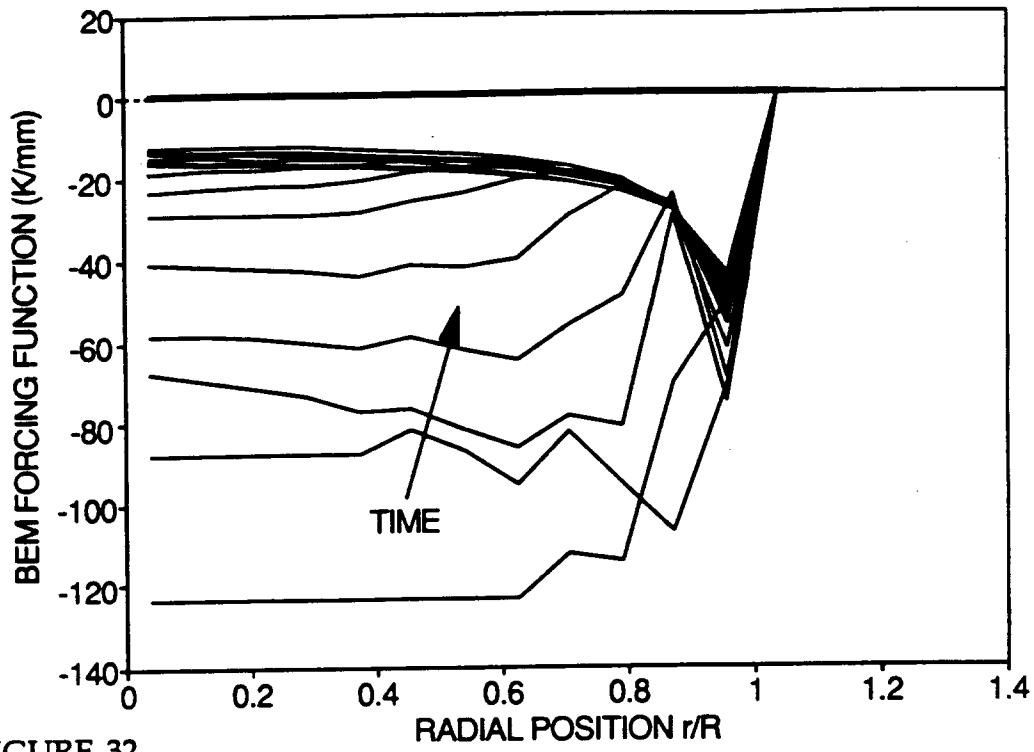

FIGURE 32
Spatial Profile of BEM Forcing Function: Run 5 (Geometry Model A)

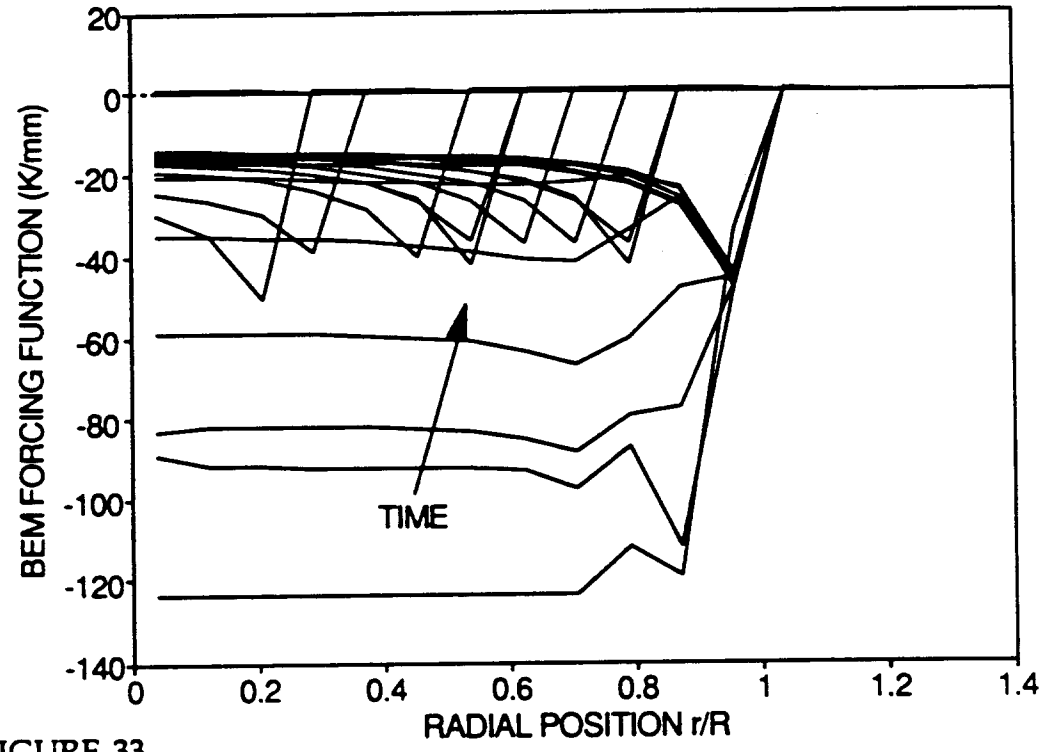

FIGURE 33
Spatial Profile of BEM Forcing Function: Run 16 (Geometry Model B)

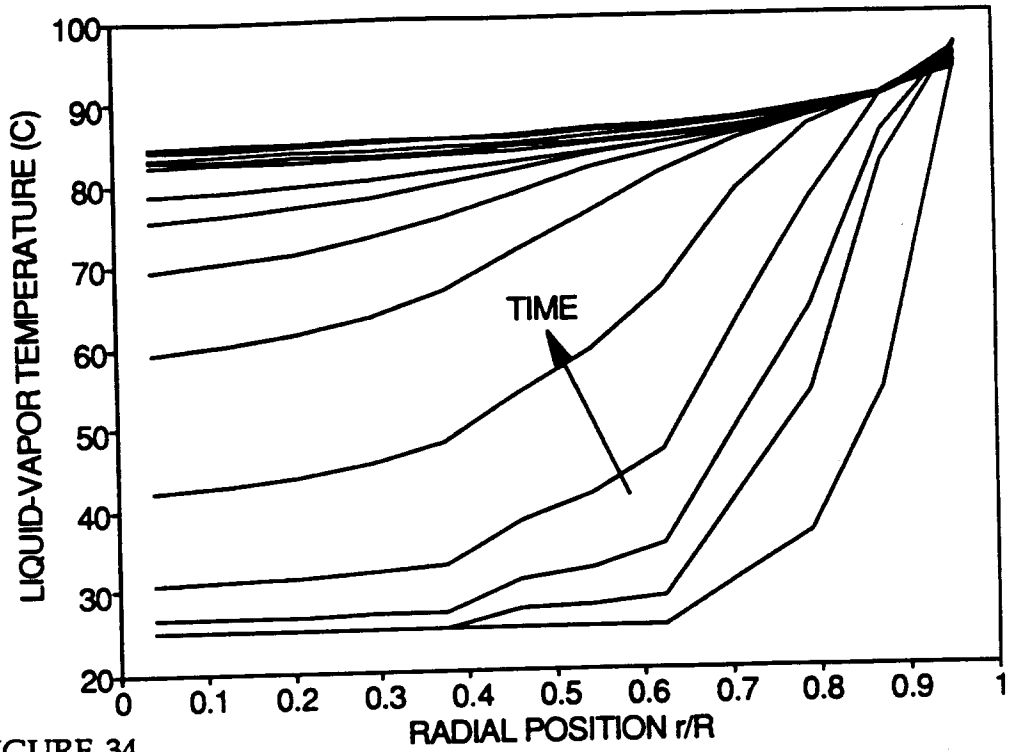

FIGURE 34
Spatial Profile of Liquid-Vapor Temperature: Run 5 (Geometry Model A)

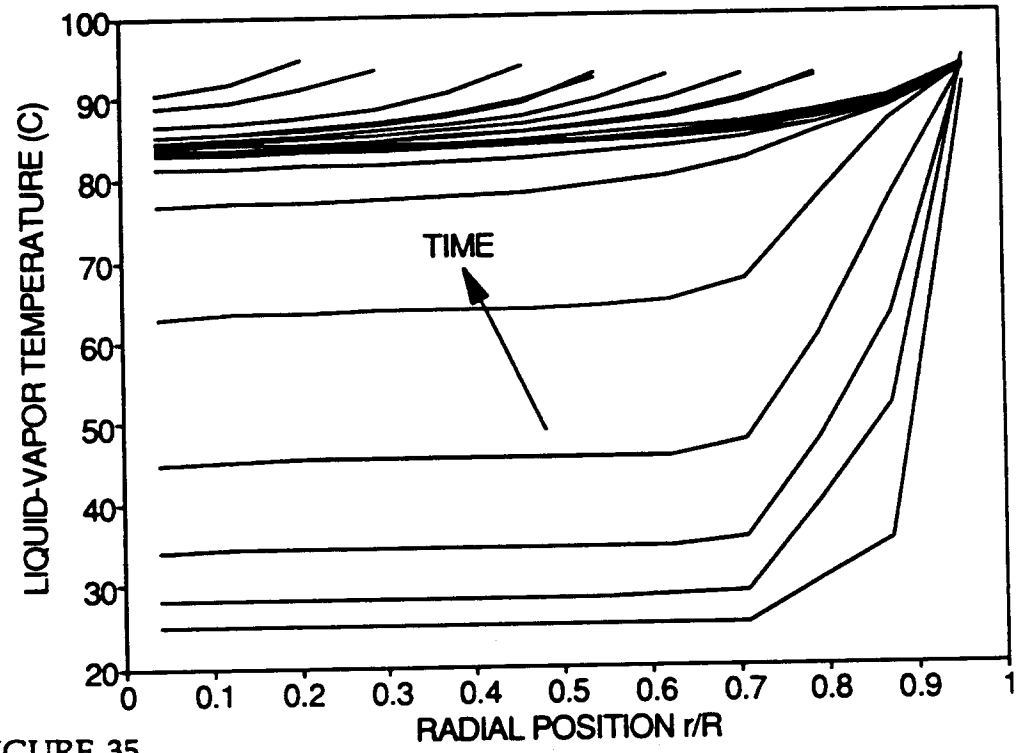

FIGURE 35
Spatial Profile of Liquid-Vapor Temperature: Run 16 (Geometry Model B)

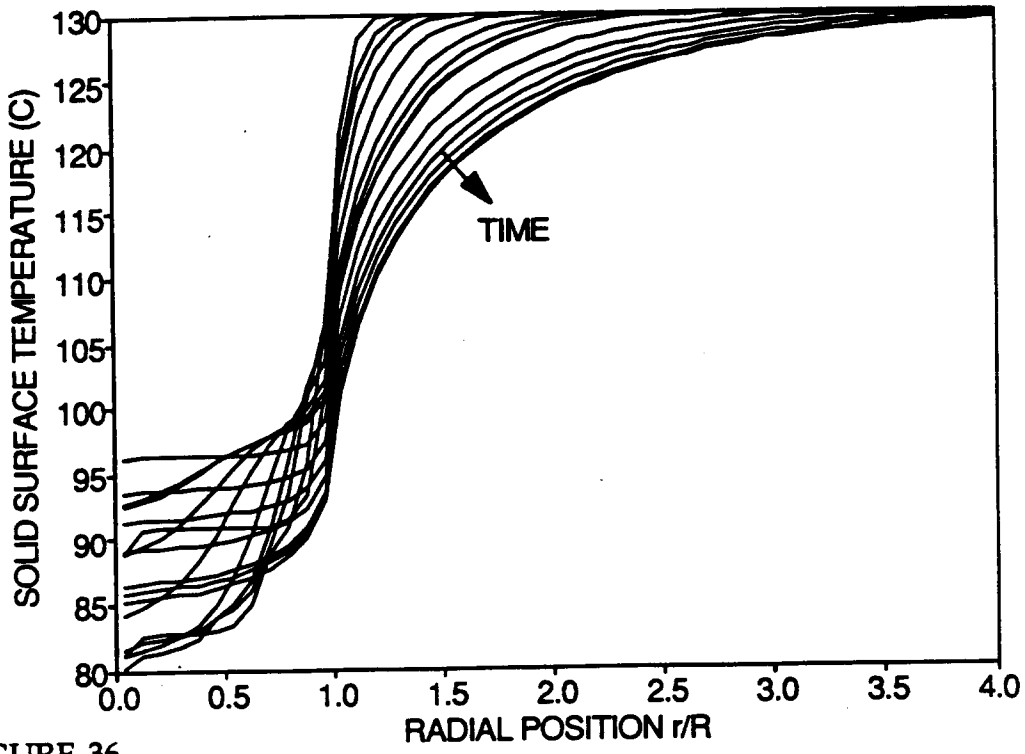

FIGURE 36
Spatial Profile of Solid Surface Temperature: Run 5 (Geometry Model A) ($t < \tau$)

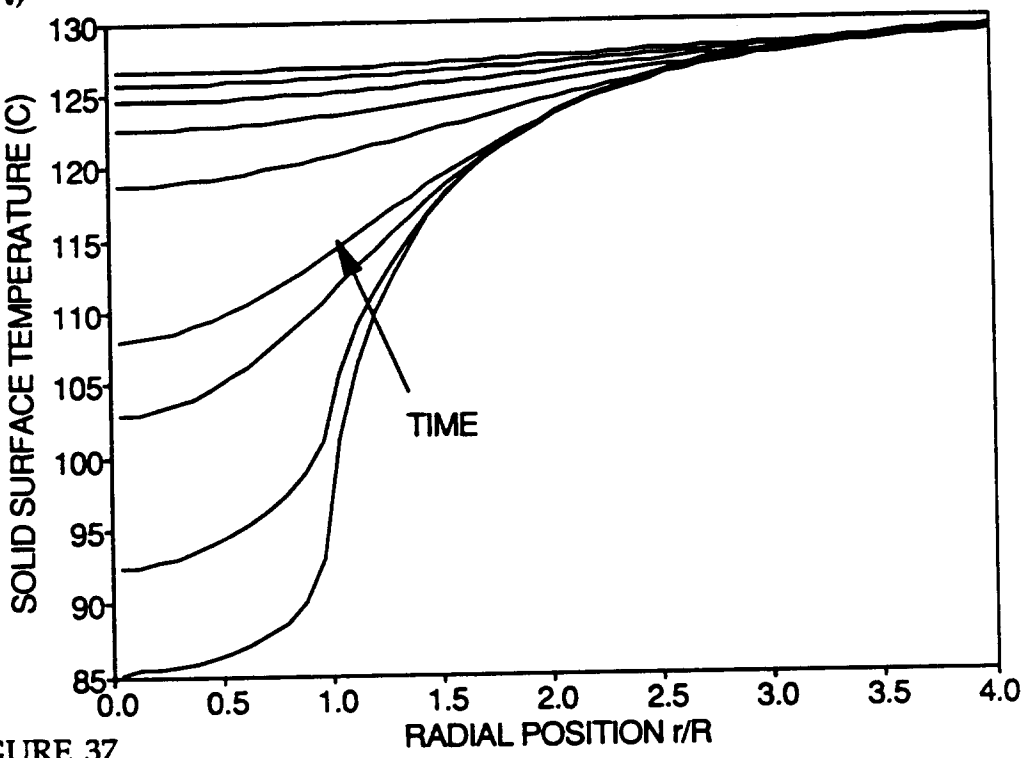

FIGURE 37
Spatial Profile of Solid Surface Temperature: Run 5 (Geometry Model A) ($t > \tau$, recovery)

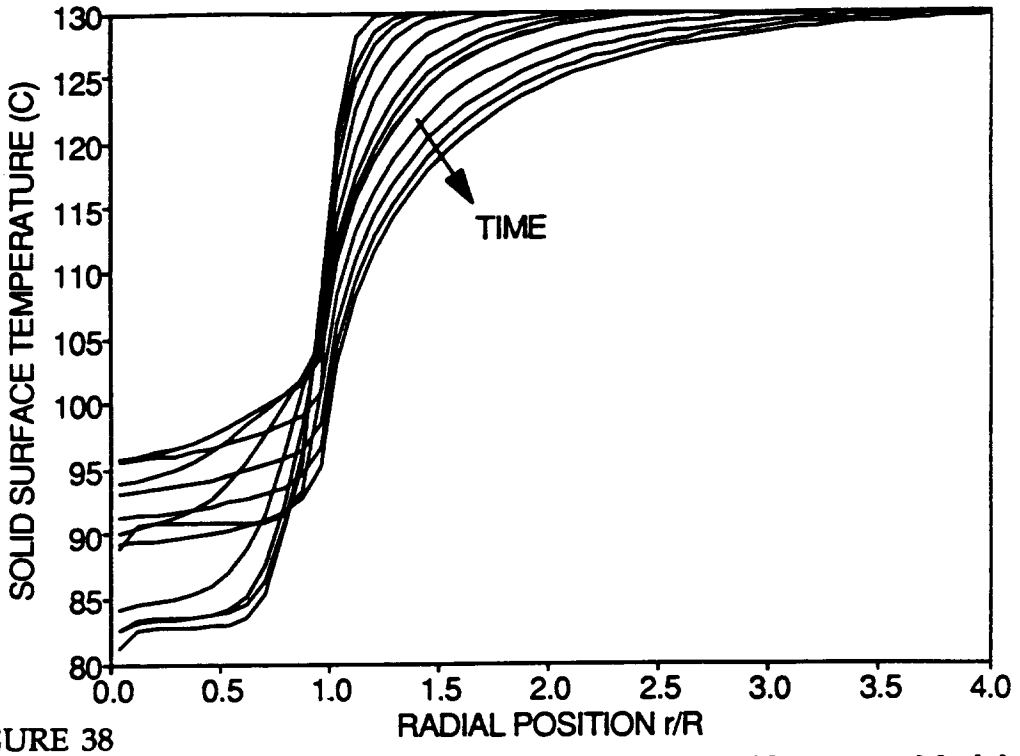

FIGURE 38
Spatial Profile of Solid Surface Temperature: Run 16 (Geometry Model B)
($t < t_{recession}$)

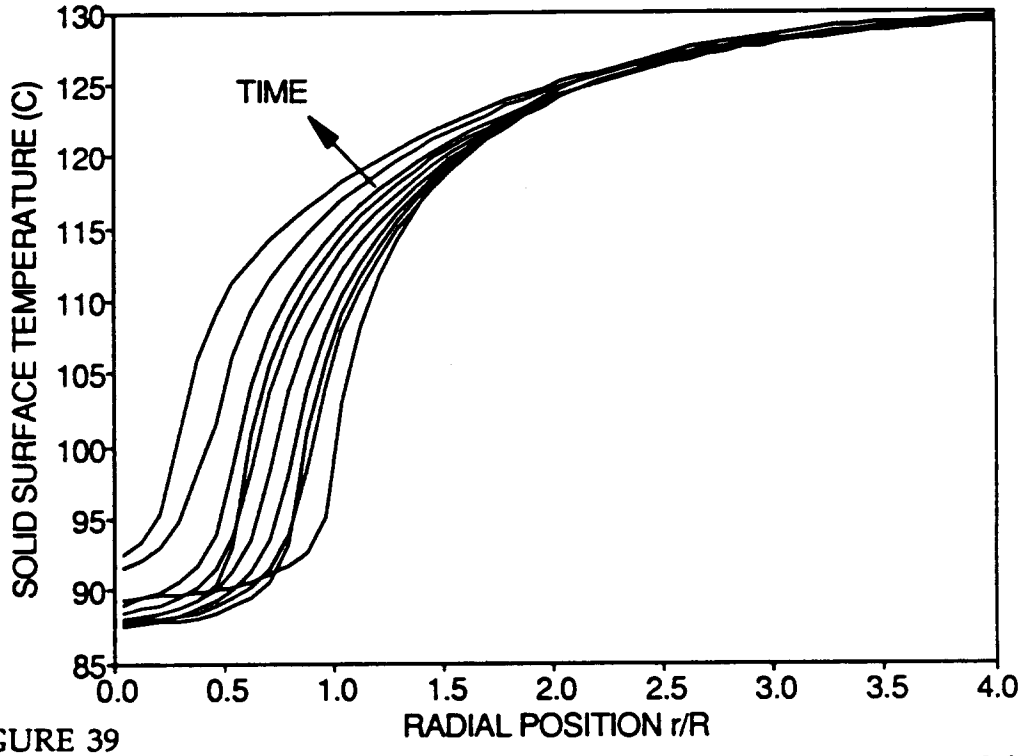

FIGURE 39
Spatial Profile of Solid Surface Temperature: Run 16 (Geometry Model B)
($t_{recession} < t < \tau$; droplet recession)

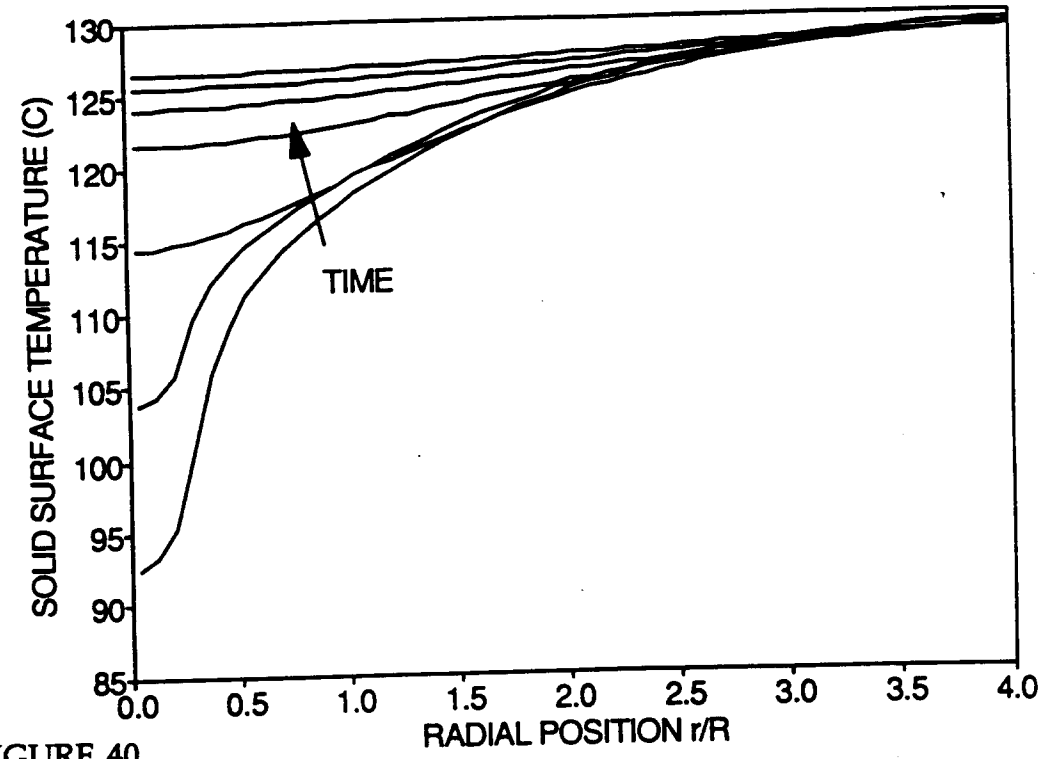

FIGURE 40
Spatial Profile of Solid Surface Temperature: Run 16 (Geometry Model B) ($t > \tau$; recovery)

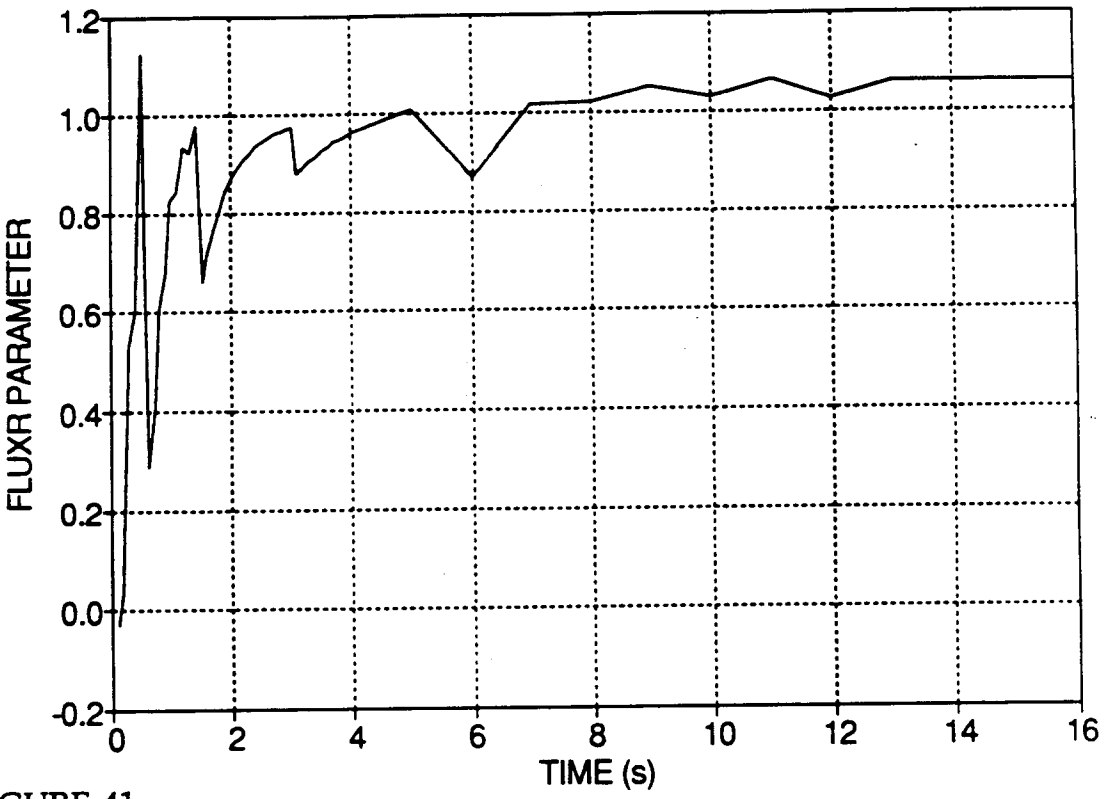

FIGURE 41
Time Dependence of FLUXR Parameter: Run 5

the thin edge and that as the entire droplet reaches quasi-steady state status, T_i becomes fairly uniform. The solid surface temperature graphs show that for the combinations of inputs used the influence of the droplet is practically restricted to a disk of radius $4R_0$. Additionally, after a short initial period, the solid surface temperature under the droplet becomes fairly uniform. This behavior was also produced in the theoretical results of Liao [12] and is due to the high fluxes at the droplet edge providing extra cooling to that annular region of the solid. Therefore, there is a straightforward relationship between solid surface flux and temperature as would be expected in a linear system. Figure 39 shows that the solid surface temperature profile has time to shift along with the droplet recession. In fact, Figures 37 and 40 show that the surface temperature begins to quickly recover after the droplet has vanished.

The transfer of liquid solution control from the transient to quasi-steady state solution is illustrated by Figure 41. Every time the quantity FLUXR reaches 0.97 the outermost transient column becomes quasi-steady state. The inner annular columns reach this status nearly at the same time because the droplet thickness is somewhat uniform there.

4.1 Effect of Shape Factor β_0

Figure 42 illustrates the effect of wetted surface area for the 10 μl droplet. The upwardly concave form for the curve indicates that the evaporation time decreases that come with increased surface area (goes with square of β_0) are not as large for large β_0. One possibility for this effect is that the nearly-constant radiation component has less time to make its contribution to droplets with lower evaporation times.

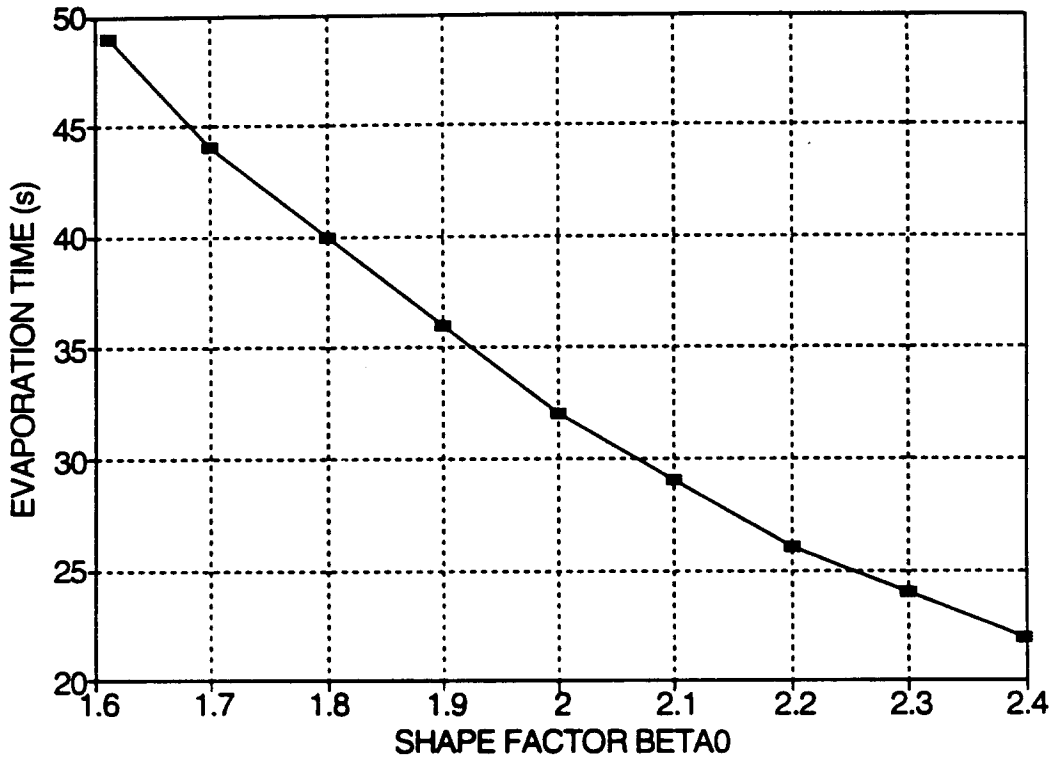

FIGURE 42
Evaporation Time vs. Geometry Factor β_0: Runs 1–9

The figures of Appendix D.1 illustrate more of the model behavior. The volume graph shows the steadiness of the dropwise evaporative process. The volume rate of change curves show how the evaporation mass transfer rates increase with liquid-vapor temperature. The nonsmooth features around 5 s for high β_0 possibly indicate numerical inaccuracies caused by the switch to the longer time interval. However, for small β_0 the switch is remarkably smooth. The droplet bottom surface spatial average flux graph demonstrates the temporal behavior of the conduction contribution. After the high initial values, the average flux becomes roughly constant. Furthermore, the behavior is almost independent of β_0, meaning that the temperature difference across the droplet and droplet thickness effects cancel each other out. Figures 60 through 68 show precisely how the droplet upper and lower surface temperatures approach each other as the droplet thins.

The recollection memory curves (Figure 69) show that solids with more compact droplets (small β_0) require less recollection time than solids with spread out droplets to know their present state. At the point that the droplet vanishes memory increases all the way back to deposition because the strong and recent forcing functions are gone and no longer overpower relatively long past events. The exponentially decaying behavior of the Green's functions makes it surprising that the memory should be as great as shown. The reason is that the Green's functions tend to level off for recollection times from 10 to 30 s and that the early forcing functions are the strongest; the effects compensate and the memory remains long.

4.2 Effect of Initial Volume V_0

The wetted area (and R_0) was kept constant in this investigation in an attempt

to learn independently the effect of the uncertainty in the experimental value of V_0. The effect was to cut τ from 32 to 28 s for a 1 μl decrease in initial volume. The relationship between V_0 and τ is nearly linear for small changes at least in this case. Appendix D.2 gives the graphical results for this small parametric study. Figure 71 shows that the thinnest droplet (smallest V_0) has only a slightly higher mass transfer rate (as would be required for the linear relationship between V_0 and τ).

4.3 Effect of Initial Temperatures

This investigation clearly shows the strong sensitivity of the solid-droplet system to initial solid surface temperature. Raising T_{S0} 20°C from 120 to 140°C cut the evaporation time by almost 40%. The system is far less sensitive to changes in the initial liquid temperature and radiant panel temperature. This can be explained in that the sensible energy change in the droplet and the radiant contribution are relatively small parts of the required energy. Quantitative outputs are found in Appendix D.3. As expected the droplet initially at 60°C takes less time to get up to the maximum mass transfer rate than its 20°C counterpart.

4.4 Effect of Geometry Model

Implementation of geometry Model B produced some very interesting results, found in Appendix D.4. Figures 82 and 83 show that Model B results in longer evaporation times not because of the larger value of θ_0 but because of the slow down in volume flux after droplet recession. In fact, Figure 84 shows that time to reaching the receding contact angle is nearly independent of initial contact angle. The similar plot with time scaled to τ (Figure 85) shows that the effect of increasing β_0 is to reach recession comparatively

faster. The volume plot (Figure 82) suggests that the special cylinder-geometry-model droplet would have the longest evaporation time of the β_0 = 2.0 droplets if it had a receding property. This coupled with the observed order of the curves leads to the conclusion that evaporation time increases slightly with θ_0, all other quantities equal. Severe local thinning at the edge, where area is greatest, is more efficient at pushing flux through the wetted area than moderate thinning throughout the droplet.

4.5 Comparison with Experimental Results

Comparisons between the theoretical and experimental results can be made in two ways: through the evaporation time and through the radial and temporal solid surface temperature dependence.

Evaporation Time Comparison

If one accepts all the values given by Dawson [13], then geometry Model A predicts an evaporation time 33% too long and Model B 58% too long. However, there are several reasonable explanations that the source(s) of the discrepancy may be related to the experiment (possible sources of discrepancy related to the code are discussed in Chapter 5):

(1) Kidder [8] performed experiments very similar to Dawson's and measured an average τ of 31.2 s for an average β_0 of 2.0. This is very close to the 32 s of Model A. Kidder's radiant panels were of a different geometry, but this effect can be argued not to be significant for several reasons: a) Chapter 2 showed that the reflectivity dependence on incident angle cannot be expected to change F_ε much; b) overall energy balances and equal T_{S0} values mean that the radiative fluxes should be

close; and c) the spectral dependence of Kidder's incident radiation (T_R value) would not be too different from Dawson's. Then, the simplified radiation model predicts very similar droplet behavior.

(2) β_0 was difficult to experimentally measure in Dawson's setup. Also radiation-induced surface tension relaxation at the droplet edge may mean that R_0 was really larger than it seemed.

(3) Slight errors in the measurement of V_0 or T_{S0} could have significant effect on evaporation time.

(4) Dawson measured evaporation time by marking the time at which the video tape showed the temperature profile begin to snap back toward the initial conditions. Results using geometry Model B showed that the snapping back of the temperature profile would begin at the point of recession not complete evaporation. Figure 82 shows that this effect may be on the order of 4 or 5 s.

Solid Surface Temperature Profile Comparison

This comparison is greatly complicated by the difference in evaporation times and uncertainty with β_0. The $\beta_0 = 2.0$, Model A theoretical results were used for comparison because of the agreement with Kidder's value of β_0. Discrepancies in β_0 values cause a scaling problem with the experimental data, which is given as a function of r/R_0. The purpose of Figures 43 and 44, which show the temperature profiles overlayed, is only to demonstrate the agreement of the general behavior of the theoretical and experimental T_S.

4.6 The Constant Heat Flux Approximation

The spatial and temporal behavior of the solid surface fluxes suggest that the classical constant heat flux solution might be a useful model of the

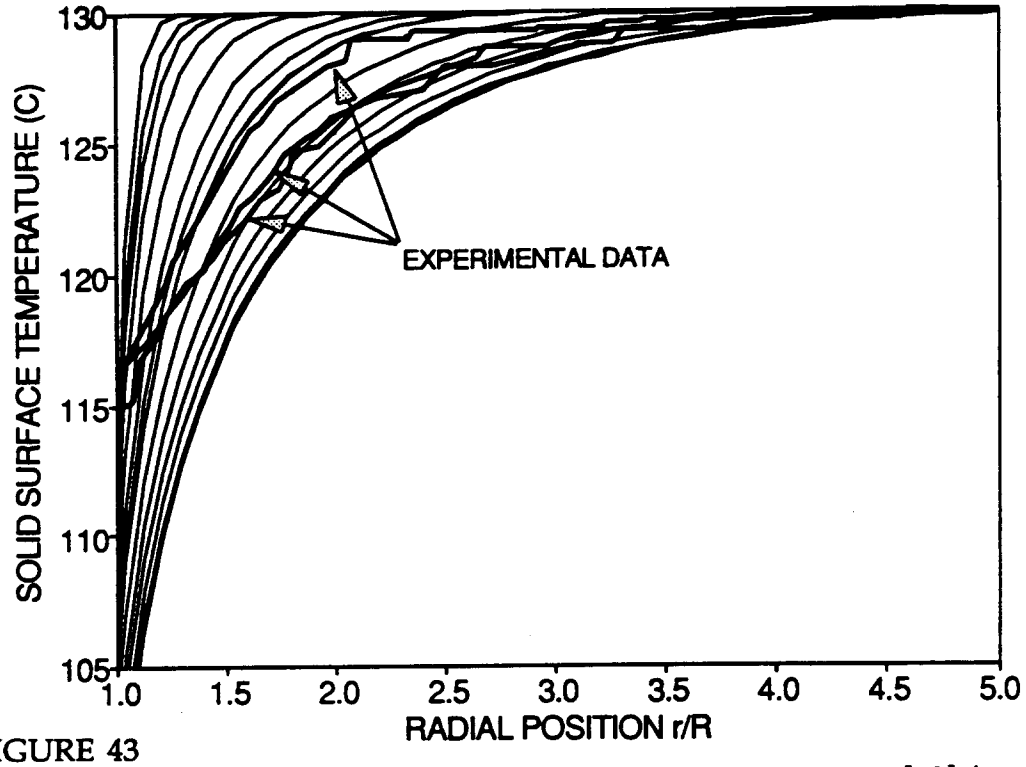

FIGURE 43
Run 5 Temperature Profile Compared with Experimental Data [13] ($t < \tau$)

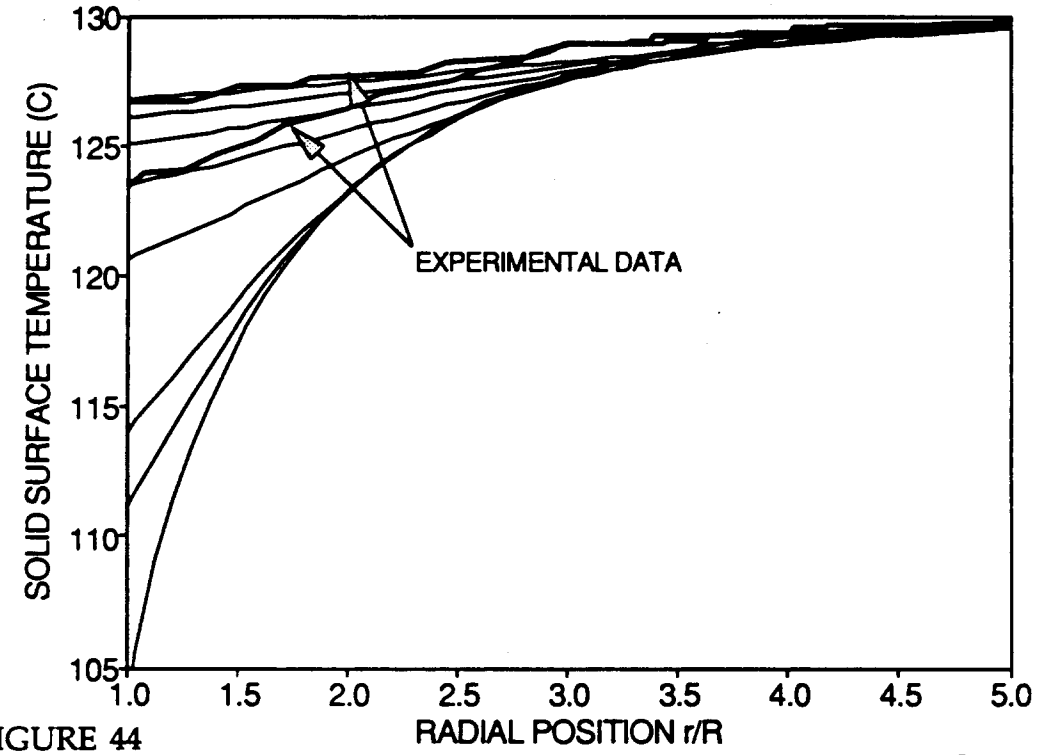

FIGURE 44
Run 5 Temperature Profile Compared with Experimental Data [13]
($t > \tau$; recovery)

evaporative process. Carslaw and Jaeger [29] and later Beck, et al., [20] showed that for a semi-infinite solid surface suddenly subjected to a constant and uniform heat flux q out of solid over a circular region of radius R with no other heat transfer on the remainder of the surface:

$$T_{S0} - T(r, 0, t) = (qR/k_s) \int_0^\infty J_0(\beta r/R)\, J_1(\beta)\, (1/\beta)\, \text{erf}\, [\beta(\alpha_s t)^{1/2}/R]\, d\beta \quad (4.1)$$

This expression can be developed using the Green's function solution equation (GFSE) and judicious analytical integration. The integration on β originates from the particular Green's function expression most amenable to the problem. This solution has no provision for the case when the droplet is evaporated and the surface temperature begins to recover.

The author overcame this limitation by changing the limits on the recollection time integral of the GFSE from (0, t) to (t - τ, t) where t - τ is the time since the flux disk was suddenly removed. After lengthy manipulations the solution valid for times before and after disk removal is obtained:

$$T_{S0} - T(r, 0, t) =$$
$$(qR/k_s) \int_0^\infty J_0(\beta r/R)\, J_1(\beta)\, (1/\beta)\, \{\text{erf}\, [\beta(\alpha_s t)^{1/2}/R] - \text{erf}\, [\beta(\alpha_s t^+)^{1/2}/R]\}\, d\beta$$

where $\quad t^+ = 0 \quad$ for $t \le \tau$

$\quad\quad\quad t^+ = t - \tau \quad$ for $t > \tau$ $\quad (4.2)$

Note that as t approaches infinity both error functions approach 1 and cancel each other so that the initial temperature T_{S0} is restored.

4.7 Comparison with the Constant Heat Flux Model

Because there is an initial flux in the droplet-solid system, the value q in (4.2) must be defined to include the effect of subtracting out the initial conditions. Recall that the full GFSE includes a term for initial conditions that can be ignored only by mapping to another domain. The required definition for q is

$$q = q_{disk} - q_0 \quad \text{where } q_{disk} > 0 \text{ and } q_0 < 0 \text{ for the radiation case} \qquad (4.3)$$

The initial flux q_0 is obtained by an overall energy balance. The conductive flux q_{disk} is an output of EVAP that may be curve fit for conditions of interest to use in (4.3) and (4.2). Alternatively, the theoretical or experimental evaporation times can be curve fit, and an energy balance around the droplet used to find q_{disk}:

$$q_{disk}\,\pi R^2 \tau + q_{rad}\,\pi R^2 \tau - q_{conv}\,\pi R^2 \tau = \rho_l V_0 \Delta h$$

$$q_{disk} = \rho_l V_0 \Delta h \,/\, \pi R^2 \tau + q_{conv} - q_{rad} \qquad (4.4)$$

$$\text{where} \qquad \Delta h = c_p \Delta T + \Lambda$$

and ΔT is a representative value of the droplet temperature change before evaporation.

The computer code needed to implement (4.2) is given in Appendix E. The subroutine QAGI is called to a system library in order to perform the difficult semi-infinite numerical integration. The argument is a bessel-type damped oscillation which is difficult to integrate conventionally. In addition, the subroutine requires the code to be compiled in double precision mode.

Typical results are given in Figures 45 and 46. The curves compare very favorably with the EVAP theoretical and experimental data. Note that the temperature profile plateaus under the droplet are curved for the constant flux model. This is expected because the high edge fluxes are brought down to the average. The inability of the model to duplicate the behavior of droplet recession, especially for high β_0 (early recession), is a weakness. However, an additional curve fit to theoretical or experimental data by mapping t to some function of t in (4.2), i.e. replacing t with f(t), could improve the model validity. Another approach is to superimpose two concentric disks of variable relative strengths or areas. Use of a validated constant flux model is the current approach for a multi-droplet code in this line of research.

4.8 Use of the Constant Heat Flux Model to Test the BEM

As was mentioned in the subsection on the BEM, the constant heat flux model was used to validate the BEM section of EVAP. The procedure was simply to redefine the forcing function to be 1 for $r < 1$ and 0 for $r > 1$. All other quantities were also nondimensionalized. The nondimensional form of (4.2) was also computed. The nondimensional temperature profiles are compared at three times in Figure 47. The maximum deviation is about 1.5% of the full-scale temperature and occurs around an r value of 0.5. This test says nothing directly about how adequate the BEM grid size is for $r > R_0$ because the forcing function is zero there. However, the forcing function is normally small for $r > R_0$, and the grid size is $R_0/12$ for $r < 4R_0$. Therefore, the author concludes that the BEM section of EVAP is functionally well.

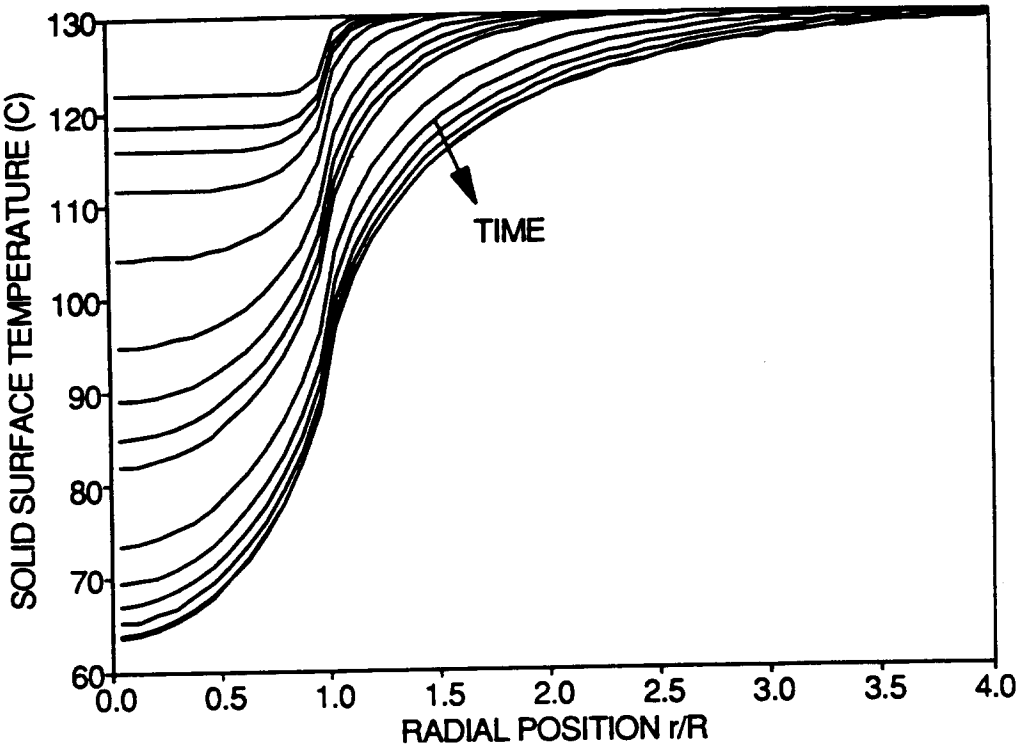

FIGURE 45
Constant Heat Flux Model ($t < \tau$)

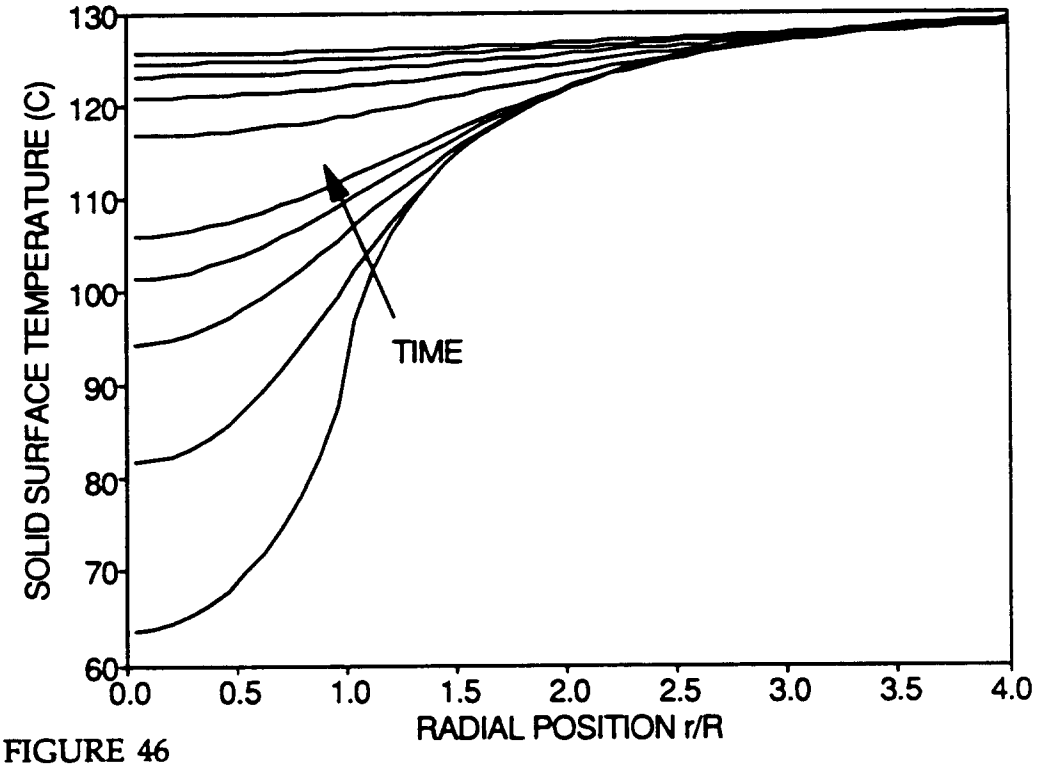

FIGURE 46
Constant Heat Flux Model ($t > \tau$; recovery)

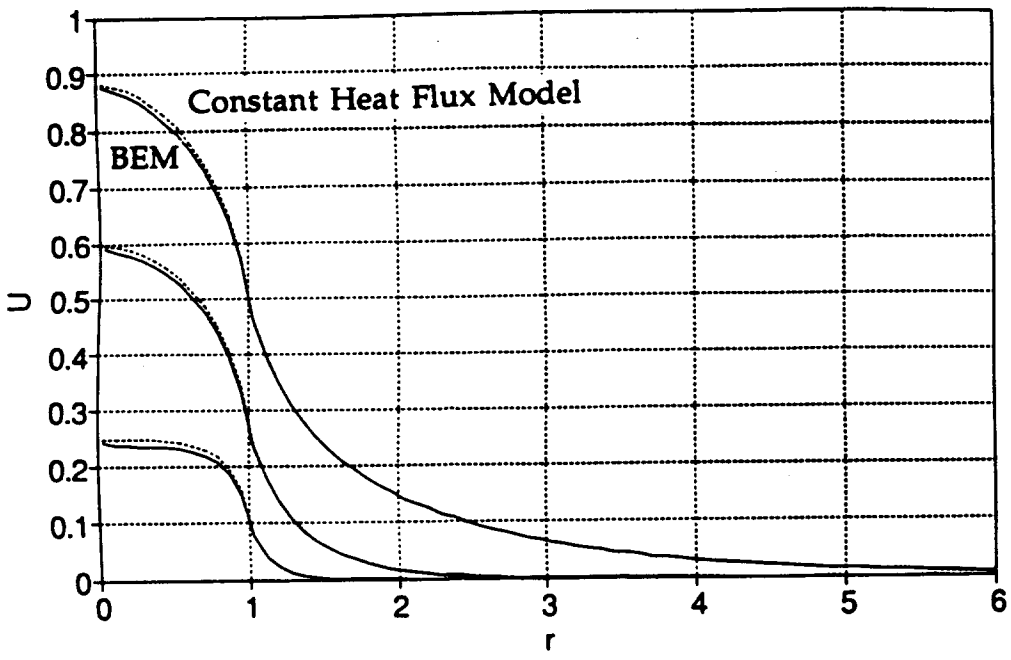

FIGURE 47
BEM Section Validation Using Constant Heat Flux Model

5. CONCLUSIONS AND RECOMMENDATIONS

The code output clearly demonstrates that the program is functionally smoothly with no numerical glitches. However, some further experimentation may help to shed light on the experimental deviations. Other combinations of T_{S0}, T_R, β_0, and grid size will surely help. It is improbable that the radiation model is the source of the deviation because of its relatively minor role for the T_{S0} = 130°C case; increasing the radiative flux 50% in Run 13 did not change the evaporation time. The geometry models are always questionable because they are empirical simplifications. Attempting to validate the program using Kidder's data [8] may be the most straightforward approach. The code was written to ensure that future researchers could carry on the work without a steep learning curve.

Although this effect did not appear in the results given in Chapter 4, the EVAP code has been observed to grossly overestimate the wetted region heat flux for the last time step with evaporation. This is possible if the droplet vanishes in the early portion of this time step. Because the droplet is very thin in this case, the calculated flux is very high and not representative of the entire time step. The remedy is to replace the false high flux with the uniform flux that would be needed to evaporate the remaining liquid over the entire time step. The following should be inserted immediately above the updating of the droplet volume:

```
IF (V .LT. DV) THEN
   DO 185 J = 1,NUMCOL
      IF (DEL(J) .GT. 0.)
```

```
     +        FLUX(J) = V * RHOH2O * HFG(SSTI(J)) / (PI * RADIUS**2 * DT)
185       CONTINUE
          END IF
```

The expression for the constant heat flux model helps to explain one difference between the cases of heating by conduction from below and by radiation from above. For equal but opposite initial fluxes q_0, the q quantity of (4.3) has a value $2q_0$ larger for the radiation case. Thus the temperature depression is greater for the radiation case, and the temperature at the solid-liquid interface can be expected to be lower. Another point of view is that the heat flux has the more difficult path to go from the solid surface down, radially in, and then up into the droplet. In the conductive case the path is straight up into the droplet. The effect of direct radiation to the droplet and radiation-induced surface tension relaxation of the droplet (increased β) counteract this enhanced temperature depression effect. The result of these compensating effects is rather similar evaporation times.

Future uses for the code with minor modifications include the following:

(1) Calculate the H(z) and F(z) functions for T_R values beyond those easily possible in the laboratory using the radiation model code. Pick nonzero H_c values and investigate the effect of high temperature incident radiation. The magnitude of the radiative flux can be kept reasonable by using small view factors.

(2) Radically alter the liquid material properties in order to test the cooling behavior of other liquids or hypothetical materials.

(3) Investigate in detail the effect of the solid properties (thermal conductivity and diffusivity).

Appendix A QuickBASIC Code for Radiation Model

```
REM Program FRAD.BAS calculates the Radiative Flux Frad
REM as a function of z (distance to surface) given the file FTHETA.DAT
REM for ftheta using the trapezoidal rule of double integration
CLS
DEFDBL A-H, O-Z
CONST pi = 3.14159265358979#
DIM NumLam(1 TO 3)                    'Highest Number for Jth Segment of Integral
DIM dlam(1 TO 3)                      'wavelength step size in um for Jth Segment

numloop = 0

INPUT "Enter the Coil Temperature Tcoil (C):    ", Tcoil
INPUT "Enter the local distance from the surface z (mm):    ", z
Tcoil = Tcoil + 273.15#
Cn = 1.33#
C1 = 374200000#
C2 = 14390#

REM Read in ftheta values
DIM fdata(0 TO 900), thetadata(0 TO 900)
OPEN "A:\FTHETA.DAT" FOR INPUT AS #33
FOR K = 0 TO 900
  INPUT #33, thetadata(K), fdata(K)
NEXT K
CLOSE #33

dtheta = pi * .1# / 180#
NumLam(1) = 16: NumLam(2) = 20: NumLam(3) = 10
dlam(1) = .05#: dlam(2) = .2#: dlam(3) = .5#

SUM3 = 0#
FOR J = 1 TO 3                        'Three Integration Segments are Necessary
  SUM2 = 0#
  FOR Ilam = 0 TO NumLam(J)
    SUM = 0#
    READ wl, alam
    tau = alam * (z / 10#)            'z in mm, alam in cm^-1
    E = C1 / (wl ^ 5# * (EXP(C2 / (wl * Tcoil)) - 1))
    PRINT "E = "; E, "wl = "; wl, "tau = "; tau

    SUM = SUM + 0#                                          'First Value is 0
    FOR thetad = .1# TO 89.95# STEP .1#
      theta = pi * thetad / 180#

      REM Retrieve ftheta
      K = INT(thetad * 10 + .5)
      ftheta = fdata(K)
      thetainput = thetadata(K)

      REM Calculate rho
      u = (Cn ^ 2# - SIN(theta) ^ 2#) ^ .5#
      Rperp = ((COS(theta) - u) / (COS(theta) + u)) ^ 2#
      Rpar = ((Cn ^ 2# * COS(theta) - u) / (Cn ^ 2# * COS(theta) + u)) ^ 2#
      rho = (Rperp + Rpar) / 2#

      REM Calculate mu (cmu here)
      A = .75# * SIN(theta)
      cmu = COS(ATN(A / (1 - A ^ 2#) ^ .5#))

      PRINT thetad, thetainput, ftheta, K, rho, cmu
      funct = ftheta * COS(theta) * SIN(theta) * (1# - rho) * EXP(-tau / cmu)
      SUM = SUM + 2# * funct
```

```
    NEXT thetad                                          'Last Value is 0
    SUM = SUM + 0#                                       'E not included in funct
    SUM = SUM * E
    PRINT "************************"
    PRINT "SUM = "; SUM, "SUM2 = "; SUM2, "SUM3 = "; SUM3
    PRINT "************************"
    PRINT "************************ numloop: NEXT Idelay
    FOR Idelay = 1 TO numloop: NEXT Idelay
    IF Ilam > 0 AND Ilam < NumLam(J) THEN
      SUM2 = SUM2 + SUM * dtheta
    ELSE
      SUM2 = SUM2 + SUM * dtheta / 2#
    END IF
  NEXT Ilam
  PRINT "+++++++++++++++++++++++++++++++++++++++"
  PRINT "SUM2 = "; SUM2
  PRINT "+++++++++++++++++++++++++++++++++++++++"
  FOR Idelay = 1 TO 2 * numloop: NEXT Idelay
  SUM3 = SUM3 + SUM2 * dlam(J) / 2#
NEXT J

PRINT "SUM3 = "; SUM3
PRINT

Frad = 2# * SUM3 / 100#
PRINT "Frad = "; Frad; " kW/m^2"
PRINT "z = "; z; " mm"
PRINT "Tcoil = "; Tcoil; " C"

REM Absorption coefficient of water alam (cm^-1) as function of wavelength (um)
DATA .20, .0691
DATA .25, .0168
DATA .30, .0067
DATA .35, .0023
DATA .40, .00058
DATA .45, .00029
DATA .50, .00025
DATA .55, .000045
DATA .60, .0023
DATA .65, .0032
DATA .70, .0060
DATA .75, .0261
DATA .80, .0196
DATA .85, .0433
DATA .90, .0679
DATA .95, .388
DATA 1.0, .363

DATA 1.0,       .363
DATA 1.2,      1.04
DATA 1.4,     12.4
DATA 1.6,      6.72
DATA 1.8,      8.03
DATA 2.0,     69.1
DATA 2.2,     16.5
DATA 2.4,     50.1
DATA 2.6,    153
DATA 2.8,   5160
DATA 3.0,  11400
DATA 3.2,   3630
DATA 3.4,    721
DATA 3.6,    180
DATA 3.8,    112
```

```
DATA 4.0,    145
DATA 4.2,    206
DATA 4.4,    294
DATA 4.6,    402
DATA 4.8,    393
DATA 5.0,    312

DATA 5.0,    312
DATA 5.5,    265
DATA 6.0,   2240
DATA 6.5,    758
DATA 7.0,    574
DATA 7.5,    546
DATA 8.0,    539
DATA 8.5,    543
DATA 9.0,    557
DATA 9.5,    587
DATA 10.0,   638
```

```
REM Program HRAD.BAS calculates the Volumetric Heat Generation Rate Hrad
REM as a function of z (distance to surface) given the file FTHETA.DAT
REM for ftheta using the trapezoidal rule of double integration
CLS
DEFDBL A-H, O-Z
CONST pi = 3.14159265358979#
DIM NumLam(1 TO 3)                    'Highest Number for Jth Segment of Integral
DIM dlam(1 TO 3)                      'wavelength step size in um for Jth Segment

numloop = 0

INPUT "Enter the Coil Temperature Tcoil (C):    ", Tcoil
INPUT "Enter the local distance from the surface z (mm):    ", z
Tcoil = Tcoil + 273.15#
Cn = 1.33#
C1 = 374200000#
C2 = 14390#

REM Read in ftheta values
DIM fdata(0 TO 900), thetadata(0 TO 900)
OPEN "A:\FTHETA.DAT" FOR INPUT AS #33
FOR K = 0 TO 900
  INPUT #33, thetadata(K), fdata(K)
NEXT K
CLOSE #33

dtheta = pi * .1# / 180#
NumLam(1) = 16: NumLam(2) = 20: NumLam(3) = 10
dlam(1) = .05#: dlam(2) = .2#: dlam(3) = .5#

SUM3 = 0#
FOR J = 1 TO 3                        'Three Integration Segments are Necessary
  SUM2 = 0#
  FOR Ilam = 0 TO NumLam(J)
    SUM = 0#
    READ wl, alam
    tau = alam * (z / 10#)            'z in mm, alam in cm^-1
    E = C1 / (wl ^ 5# * (EXP(C2 / (wl * Tcoil)) - 1))
    PRINT "E = "; E, "wl = "; wl, "tau = "; tau

    SUM = SUM + 0#                                    'First Value is 0
    FOR thetad = .1# TO 89.95# STEP .1#
      theta = pi * thetad / 180#

      REM Retrieve ftheta
      K = INT(thetad * 10 + .5)
      ftheta = fdata(K)
      thetainput = thetadata(K)

      REM Calculate rho
      u = (Cn ^ 2# - SIN(theta) ^ 2#) ^ .5#
      Rperp = ((COS(theta) - u) / (COS(theta) + u)) ^ 2#
      Rpar = ((Cn ^ 2# * COS(theta) - u) / (Cn ^ 2# * COS(theta) + u)) ^ 2#
      rho = (Rperp + Rpar) / 2#

      REM Calculate mu (cmu here)
      A = .75# * SIN(theta)
      cmu = COS(ATN(A / (1 - A ^ 2#) ^ .5#))

      PRINT thetad, thetainput, ftheta, K, rho, cmu
      funct = ftheta * COS(theta) * SIN(theta) * (1# - rho) * EXP(-tau / cmu) /
 cmu
```

```
      SUM = SUM + 2# * funct
    NEXT thetad
    SUM = SUM + 0#                                  'Last Value is 0
    SUM = SUM * E * alam                            'E not included in funct
    PRINT "************************"
    PRINT "SUM = "; SUM, "SUM2 = "; SUM2, "SUM3 = "; SUM3
    PRINT "************************"
    FOR Idelay = 1 TO numloop: NEXT Idelay
    IF Ilam > 0 AND Ilam < NumLam(J) THEN
      SUM2 = SUM2 + SUM * dtheta
    ELSE
      SUM2 = SUM2 + SUM * dtheta / 2#
    END IF
  NEXT Ilam
  PRINT "++++++++++++++++++++++++++++++++++++++"
  PRINT "SUM2 = "; SUM2
  PRINT "++++++++++++++++++++++++++++++++++++++"
  FOR Idelay = 1 TO 2 * numloop: NEXT Idelay
  SUM3 = SUM3 + SUM2 * dlam(J) / 2#
NEXT J

PRINT "SUM3 = "; SUM3
PRINT

Hrad = 2# * SUM3
PRINT "Hrad = "; Hrad; " kW/m^3"
PRINT "z = "; z; " mm"
PRINT "Tcoil = "; Tcoil; " C"

REM Absorption coefficient of water alam (cm^-1) as function of wavelength (um)
DATA .20, .0691
DATA .25, .0168
DATA .30, .0067
DATA .35, .0023
DATA .40, .00058
DATA .45, .00029
DATA .50, .00025
DATA .55, .000045
DATA .60, .0023
DATA .65, .0032
DATA .70, .0060
DATA .75, .0261
DATA .80, .0196
DATA .85, .0433
DATA .90, .0679
DATA .95, .388
DATA 1.0, .363

DATA 1.0,      .363
DATA 1.2,     1.04
DATA 1.4,     12.4
DATA 1.6,     6.72
DATA 1.8,     8.03
DATA 2.0,     69.1
DATA 2.2,     16.5
DATA 2.4,     50.1
DATA 2.6,     153
DATA 2.8,    5160
DATA 3.0,   11400
DATA 3.2,    3630
DATA 3.4,     721
DATA 3.6,     180
```

```
DATA 3.8,   112
DATA 4.0,   145
DATA 4.2,   206
DATA 4.4,   294
DATA 4.6,   402
DATA 4.8,   393
DATA 5.0,   312

DATA 5.0,   312
DATA 5.5,   265
DATA 6.0,  2240
DATA 6.5,   758
DATA 7.0,   574
DATA 7.5,   546
DATA 8.0,   539
DATA 8.5,   543
DATA 9.0,   557
DATA 9.5,   587
DATA 10.0,  638
```

Appendix B FORTRAN Code for Single-Droplet Model

```
C     GLENN WHITE
C     (301) 405-5334
C     ADVISOR: MARINO DIMARZO  (301) 405-5257
C     MASTER'S THESIS -- UNIVERSITY OF MARYLAND AT COLLEGE PARK
C     EVAPORATIVE COOLING WITH RADIANT HEAT INPUT
C     PROGRAM TO CALCULATE TRANSIENT SURFACE TEMPERATURE DISTRIBUTION
C     WRITTEN FOR NIST CENTER FOR FIRE RESEARCH
C
C234567
      PROGRAM EVAP01
C.....DEFINE COMMON BLOCK TO ALLOW SUBROUTINE UPGEOB TO KEEP VALUES OF CERTAIN VARIABLES STATIC
      REAL BVAR1, BVAR2, BVAR3, BVAR4, BVAR5, BVAR6, BVAR7, BVAR8, BVAR9, BVAR10
      REAL BVAR11, BVAR12, BVAR13, BVAR14, BVAR15
      COMMON /GEOBBL/ BVAR1, BVAR2, BVAR3, BVAR4, BVAR5, BVAR6, BVAR7, BVAR8,
     *                BVAR9, BVAR10, BVAR11, BVAR12, BVAR13, BVAR14, BVAR15
C.....PARAMETERS WHICH MAY BE SET HERE BUT ARE NOW INPUT AS VARIABLES THROUGH THE MONITOR.....
C.....THE TIME TO EXIT THE TIME STEP LOOP TEND IN S
C      PARAMETER (TEND = 60.)
C.....THE INITIAL VOLUME V0 IN M**3
C      PARAMETER (V0 = 10E-09)
C.....THE INITIAL SHAPE PARAMETER BETA0
C      PARAMETER (BETA0 = 2.00)
C.....THE GEOMETRY MODEL DESIRED, 'A' OR 'B'
C      PARAMETER (GEOMOD = 'B')
C.....THE INITIAL CONTACT ANGLE THETA0 IN DEGREES (>THETAA) (GEOMOD = 'B' ONLY)
C      PARAMETER (THET0D = 60.)
C.....THE INITIAL SURFACE TEMPERATURE (TS0)
C      PARAMETER (TS0 = 130.)
C.....THE RADIATIVE HEATER COIL TEMPERATURE IN C (475-650)
C      PARAMETER (TCOIL = 510.)
C.....NEGLECT RATIO TO DECIDE WHEN TO TERMINATE MARCH BACK INTO TIME
C      PARAMETER (NEGRAT = 0.01)
C.....ROOT OF OUTPUT FILE NAMES (6 CHARS)
C.....FILE NAME OF OUTPUT OF MAIN DROPLET PARAMETERS IS (FILNAM//'.dat')
C.....FILE NAME OF OUTPUT OF SOLID SURFACE TEMPERATURES AND FLUXES IS (FILNAM//'.out')
C.....FILE NAME OF OUTPUT OF SPATIAL DROPLET SURFACE TEMPERATURES AND FLUXES IS (FILNAM//'.ins')
C.....FILE NAME OF OUTPUT OF SPATIAL DROPLET HEIGHT AS FUNCTION OF TIME IS (FILNAM//'.geo')
C.....FILE NAME OF OUTPUT OF CUMULATIVE ENERGY TRANSFERS IS (FILNAM//'.bal')
C      PARAMETER (FILNAM = '2run01')
C.....NUMBER OF TIMES TO OUTPUT SPATIAL TEMPERATURE AND FLUX DATA
C      PARAMETER (NDATTM = 5)

C.....SPECIFY TYPES OF THE INPUT VARIABLES DEFINED ABOVE
      REAL TEND, V0, BETA0, TS0, TCOIL, NEGRAT, THET0D
      CHARACTER*6 FILNAM
      CHARACTER*10 FILNA1, FILNA2, FILNA3, FILNA4, FILNA5
      CHARACTER*1 GEOMOD
      INTEGER NDATTM

C     PARAMETERS WHICH CAN BE ADJUSTED HERE WITHOUT ANY OTHER CHANGES ---------------------------------
C     THE RECEDING ANGLE THETAR IN DEGREES  GEOMOD = 'B' ONLY)
      PARAMETER (THETRD =  7 )
C     THE AMBIENT TEMPERATURE
      PARAMETER (TAMB = 25 )
C     THE INITIAL DROPLET TEMPERATURE
      PARAMETER (TL = 25.)
C     MINIMUM FLUX RATIO PERC TO DEFINE SS IN A COL (E G  99)
      PARAMETER (PERC = 97.)

C     PARAMETERS NOT TO BE CHANGED WITHOUT OTHER CHANGES ----------------------------------------------
C     WARNING: DO NOT ADJUST TIMES  EXCEPT TEND  WITHOUT CHANGING SUBROUTINES RECONF, WEIGHT, AND WGHT
C     THE STARTING TIME STEP DTSHRT IN S
      PARAMETER (DTSHRT = 0.1)
C     THE TIME STEP DTLONG IN S
      PARAMETER (DTLONG = 1.0)
C     THE TIME TO USE THE STARTING TIME STEP TSHORT IN S
```

```
      PARAMETER (TSHORT = 4.)
C.....NUMBER OF ANNULAR COLUMNS TO MODEL THE DROPLET
      PARAMETER (NUMCOL = 12)
C.....TOTAL NUMBER OF NODES N PER COL (DZ=DELTA/(N-2))
      PARAMETER (N = 12)
C.....THE NUMBER OF TIMES USED FOR THE BEM SUBROUTINE
      PARAMETER (NUMBET = 100)
C.....THE NUMBER OF BEM NODES USED
      PARAMETER (NUMNOD = 78)
C.....................................................................
      PARAMETER (PI = 3.14159265358979)

C.....VARIABLE TYPE SPECIFICATIONS*****************************************************
      REAL GAMMA, THETA0, THETAR, V, RAD0, RADIUS, VR, RHOH2O
      REAL KLSIS, ALFSIS, EPSF, SIGMA, F, EPSILN, FRAD0, FRADS
      REAL HR, H, HO, FLUX0, TC, DT, KS, FLUX05, TIME, TI
      REAL NFLUX, DVCOL, DV, KAVG, QC, SSFI, SSDFDT, SSA, SSB, SSTEMP
      REAL EXPR, SSTOLD, FLUXR, SSTIME, EVTIME, THTIME, MEMORY, BETA, THETA, FO
      REAL CONDUC, RADIAT, CONVEC, LATENT, MAXSEN, SENSIB
      REAL TICONV, TICAVG, TAVG, FAVG
      INTEGER I, J, K, L, NTRANC
      CHARACTER*1   TSFLAG, TRNBGN, EVFLAG, THFLAG

C.....DESCRIPTION OF VARIABLES (ALL UNITS ARE SI EXCEPT TEMPERATURE IN C):
C.....GAMMA: CRANK-NICHOLSON CONSTANT; THETA0: INITIAL DROPLET CONTACT ANGLE;
C.....THETAR: RECEDING CONTACT ANGLE (GEOM MODEL B ONLY); V: CURRENT DROPLET VOLUME;
C.....RAD0: INITIAL WETTED RADIUS; RADIUS: CURRENT WETTED RADIUS;
C.....VR: VOLUME AT WHICH RECEDING BEGINS (GEOM MODEL B ONLY);
C.....RHOH2O: DENSITY OF WATER; KLSIS: SEMI-INFINITE SOLUTION (SIS) VALUE OF CONDUCTIVITY FOR DROPLET;
C.....ALFSIS: SIS THERMAL DIFFUSIVITY VALUE OF DROPLET;
C.....EPSF: ABSORPTIVITY OF DROPLET GIVEN DISTRIBUTION OF INCIDENT RADIATION BY POLAR ANGLE;
C.....SIGMA: RADIATION CONSTANT; F: THEORETICAL TILE VIEW FACTOR;
C.....EPSILN: SOLID EMMISIVITY; FRAD0: RADIATIVE FLUX PENETRATING INTO SURFACE OF DROPLET;
C.....FRADS: RADIATIVE FLUX ABSORBED BY SOLID; HR: RE-RADIATION HEAT COEFF OF SOLID;
C.....H: CONVECTIVE HEAT COEFF; HO: OVERALL HEAT COEFF; FLUX0: INITIAL HEAT FLUX;
C.....TC: SIS CONTACT TEMPERATURE; DT: TIME STEP; KS: SOLID CONDUCTIVITY;
C.....FLUX05: SIS HEAT FLUX FOR TIME STEP 0.5; TIME: ELAPSED TIME SINCE DEPOSITION;
C.....TI: LIQUID-VAPOR INTERFACIAL TEMPERATURE; NFLUX: MOLAR FLUX LEAVING DROPLET;
C.....DVCOL: CHANGE IN VOLUME OF ANNULAR DROPLET COLUMN; DV: CHANGE IN VOLUME OF DROPLET;
C.....KAVG: AVERAGE STEADY-STATE (SS) CONDUCTIVITY; QC: CONDUCTIVE HEAT FLUX;
C.....SSFI, SSDFDT, SSA, SSB, SSTEMP, EXPR, SSTOLD: SS PARAMETERS FOR TAYLOR LINEARIZATION OF LIQUID-VAPOR BC;
C.....FLUXR: RATIO OF BOUNDARY TEMPERATURE GRADIENTS OF OUTERMOST TRANSIENT ANNULAR DROPLET COLUMN;
C.....SSTIME: TIME UNTIL ALL COLUMNS REACH SS; EVTIME: TIME UNTIL DROPLET EVAPORATES;
C.....THTIME: TIME UNTIL RECEDING ANGLE IS REACHED (GEOM MODEL B ONLY)     .
C.....MEMORY: RECOLLECTION TIME AT WHICH EFFECT OF FORCING FUNCTION BECOMES NEGLIGIBLE (BASED ON NEGRAT);
C.....BETA: CURRENT SPLAT FACTOR (BASED ON RADIUS AND SPHERE OF CURRENT VOLUME);
C.....THETA: CURRENT DROPLET CONTACT ANGLE IN RADIANS (GEOM MODELS A AND B);
C.....FO: FOURIER NUMBER ALPHA*TIME/DEL(J)^2 USED TO DETERMINE IF COLUMN SHOULD BE SS INITIALLY;
C.....CONDUC, RADIAT, CONVEC, LATENT, MAXSEN, SENSIB: CUMULATIVE ENERGIES;
C.....TICONV: LIQUID-VAPOR INTERFACIAL TEMPERATURE; TICAVG: AREA AVERAGED TICONV;
C.....TAVG: AREA AVERAGED TEMPERATURE UNDER THE DROPLET;
C.....FAVG: AREA AVERAGED HEAT FLUX UNDER THE DROPLET;
C.....I, J, K: ARRAY INDICES; NTRANC: CURRENT NUMBER OF TRANSIENT COLUMNS;
C.....TSFLAG: FLAG -- '1' = (TIME > TSHORT);
C.....TRNBGN: FLAG -- '1' = THE TRANSIENT TRIDIAGONAL SOLUTION METHOD HAS BEGUN ALREADY;
C.....EVFLAG: FLAG -- '1' = (TIME > EVTIME);
C.....THFLAG: FLAG -- '1' = (TIME > THTIME)
C.....*********************************************************************

C.....FUNCTION TYPE SPECIFICATIONS
      REAL ALPHAL, CTHETA, CPA, DFDT, FRAD, HCONV, HFG, HV, KL
      REAL LVFLUX, NH2O, SISTC, SOLVAP, XI

C.....DIMENSION VECTORS AND MATRICES*****************************************
C.....DIMENSION TRIDIAGONAL MATRICES
      REAL A(N, 4, NUMCOL), AOLD(N, 4, NUMCOL)
```

```
C.....DIMENSION STEADY-STATE TI
      REAL SSTI(NUMCOL)
C.....DIMENSION SEMI-INFINITE SOLUTION TIME VECTOR
      REAL TSIS(NUMCOL)
C.....DIMENSION GEOMETRY VECTOR OF DROPLET HEIGHT
      REAL DEL(NUMCOL)
C.....DIMENSION SURFACE FLUX VECTOR
      REAL FLUX(NUMNOD)
C.....DIMENSION LAST SURFACE FLUX VECTOR
      REAL OLDFLX(NUMNOD)
C.....DIMENSION SURFACE TEMPERATURE VECTOR
      REAL T(NUMNOD)
C.....DIMENSION THE BEM FORCING FUNCTION TO BE INTEGRATED OVER
      REAL FRCFNC(NUMBET, NUMNOD)
C.....DIMENSION THE BEM WEIGHT TENSOR USED FOR PRECALCULATED VALUES
      REAL W(NUMNOD, NUMNOD, 10)
C.....DIMENSION THE VECTOR TO STORE DATA FOR SECOND BEM CALCULATION
      REAL PREDU(NUMNOD)
C.....DIMENSION THE VECTOR OF SATURATION PRESSURES OF WATER AS FUNCTION OF TEMP.
      REAL PSAT(0:100)
C.....DIMENSION THE VECTOR OF THE HFG/VFG RATIO FOR WATER AS FUNCTION OF TEMP.
      REAL HVFG(0:100)
C.....DIMENSION THE VECTOR OF FLAGS FOR INITIALIZATION OF TEMPERATURES OF COLUMNS
      CHARACTER*1 NFTIME(NUMCOL)
C.....DIMENSION THE VECTOR OF TIMES TO OUTPUT THE DATA (.1, .2,..., 3.9, 4., 5., 6.,..., TEND)
      REAL DATTIM(50)
C.....*****************************************************************

C.....READ IN TIMES TO OUTPUT DATA (THE NUMBER OF ELEMENTS MUST MATCH THE REAL STATEMENT)
C       DATA DATTIM/ 1., 5., 10., 20., 30./

C.....READ IN PSAT TABLE
      DATA PSAT/.006109,
     + .006567, .007056, .007577, .008131, .008721,
     + .009349, .010016, .010724, .011477, .012276,
     + .013123, .014022, .014974, .015983, .017051,
     + .018181, .019376, .020640, .021975, .02339,
     + .02487,  .02645,  .02810,  .02985,  .03169,
     + .03363,  .03567,  .03782,  .04008,  .04246,
     + .04496,  .04759,  .05034,  .05324,  .05628,
     + .05947,  .06281,  .06632,  .06999,  .07384,
     + .07786,  .08208,  .08649,  .09111,  .09593,
     + .10098,  .10624,  .11175,  .11749,  .12349,
     + .12975,  .13628,  .14309,  .15019,  .15758,
     + .16529,  .17331,  .18166,  .19036,  .19940,
     + .20881,  .21860,  .22877,  .23934,  .2503,
     + .2617,   .2736,   .2859,   .2986,   .3119,
     + .3256,   .3399,   .3546,   .3699,   .3858,
     + .4022,   .4192,   .4368,   .4550,   .4739,
     + .4934,   .5136,   .5345,   .5560,   .5783,
     + .6014,   .6252,   .6498,   .6752,   .7014,
     + .7284,   .7564,   .7852,   .8149,   .8455,
     + .8771,   .9097,   .9433,   .9778,  1.01325/

C.....READ IN HVFG TABLE
      DATA HVFG/12.1264,
     + 12.9767, 13.8792, 14.8355, 15.8487, 16.9224,
     + 18.0581, 19.2596, 20.5308, 21.8733, 23.2915,
     + 24.7897, 26.3694, 28.0369, 29.7935, 31.6445,
     + 33.5952, 35.6474, 37.8062, 40.0786, 42.4658,
     + 44.9764, 47.6111, 50.3778, 53.2823, 56.3274,
     + 59.5199, 62.8685, 66.3741, 70.0449, 73.8911,
     + 77.9136, 82.1186, 86.5191, 91.1178, 95.9191,
     + 100.932, 106.171, 111.638, 117.334, 123.281,
     + 129.479, 135.931, 142.658, 149.666, 156.964,
     + 164.551, 172.439, 180.646, 189.182, 198.047,
```

```
     + 207.271, 216.823, 226.778, 237.373, 247.300,
     + 258.877, 270.378, 282.332, 294.671, 307.497,
     + 320.763, 334.509, 348.725, 363.464, 378.665,
     + 394.431, 410.756, 427.570, 444.927, 462.966,
     + 481.498, 500.734, 520.567, 540.948, 562.086,
     + 583.816, 606.207, 629.439, 653.284, 677.868,
     + 703.116, 729.257, 756.202, 783.711, 812.178,
     + 841.662, 871.739, 902.695, 934.901, 967.472,
     + 1001.597, 1035.943, 1071.804, 1108.163, 1146.068,
     + 1184.769, 1224.430, 1265.227, 1307.048, 1349.996/

C.....INPUT USER VARIABLES
      PRINT *, 'INPUT TEND IN S:'
      READ (6,*) TEND
      PRINT *, 'INPUT V0 IN MICROLITERS:'
      READ (6,*) V0
      V0 = V0 * 1E-09
      PRINT *, 'INPUT BETA0:'
      READ (6,*) BETA0
      PRINT *, 'INPUT THE GEOMETRY MODEL DESIRED, A OR B:'
      READ (6,*) GEOMOD
      IF (GEOMOD .EQ. 'B') THEN
        PRINT *, 'INPUT THE INITIAL CONTACT ANGLE THETA0 IN DEGREES:'
        PRINT *, 'IT MUST BE >= THE MODEL A INITIAL ANGLE: ', CTHETA(BETA0) * (180. /PI)
        READ (6,*) THET0D
      END IF
      PRINT *, 'INPUT TS0 IN C:'
      READ (6,*) TS0
      PRINT *, 'INPUT TCOIL IN C (475-650):'
      READ (6,*) TCOIL
      PRINT *, 'INPUT NEGRAT, RATIO USED TO STOP MARCH BACK IN TIME (e.g. 0.01):'
      READ (6,*) NEGRAT
      PRINT *, 'INPUT ROOT NAME OF OUTPUT FILES -- 6 CHARS (e.g. 2run01):'
      PRINT *, 'NAME OF MAIN DROPLET PARAMETERS IS (FILNAM.dat)'
      PRINT *, 'NAME OF SOLID SURFACE TEMPERATURES AND FLUXES IS (FILNAM.out)'
      PRINT *, 'NAME OF SPATIAL DROPLET SURFACE TEMPERATURES AND FLUXES IS (FILNAM.ins)'
      PRINT *, 'NAME OF SPATIAL DROPLET HEIGHT AS FUNCTION OF TIME IS (FILNAM.geo)'
      PRINT *, 'NAME OF CUMULATIVE ENERGY TRANSFERS IS (FILNAM.bal)'
      PRINT *, 'GIVE OUTPUT FILE NAME INSIDE SINGLE QUOTES IF IT HAS ILLEGAL CHARACTERS'
      READ (6,*) FILNAM
      PRINT *, 'INPUT THE NUMBER OF TIMES TO OUTPUT THE SPATIAL DATA (INTEGER):'
      READ (6,*) NDATTM
      PRINT *, 'INPUT THE TIMES TO OUTPUT THE SPATIAL DATA (1.,2.,...,3.9,4.,.5,....,TEND):'
      DO 99 L = 1, NDATTM
        READ (6,*) DATTIM(L)
   99 CONTINUE

C     CONCATENATE THE REQUIRED OUTPUT FILE NAMES
      FILNA1 = FILNAM//'.dat'
      FILNA2 = FILNAM//'.out'
      FILNA3 = FILNAM//'.ins'
      FILNA4 = FILNAM//'.geo'
      FILNA5 = FILNAM//'.bal'

C     OPEN FILES FOR OUTPUT OF RESULTS
      OPEN (41, FILE = FILNA1)
      OPEN (42, FILE = FILNA2)
      OPEN (43, FILE = FILNA3)
      OPEN (44, FILE = FILNA4)
      OPEN (45, FILE = FILNA5)

C     INITIALIZE FLAGS
      TSFLAG = '0'
      TEMBGN = '0'
      EVFLAG = '0'
      TMFLAG = '0'
```

```
      REMFLG = '0'

C.....CONVERT TO RADIANS
      THETA0 = (PI / 180.) * THET0D
      THETAR = (PI / 180.) * THETRD

      PRINT *, 'ASSUMING MODEL A THE INITIAL THETAA (DEG)=   ', CTHETA(BETA0) * (180. / PI)

C.....SET VOLUME TO INITIAL VOLUME
      V = V0
C.....CALCULATE THE INITIAL RADIUS
      RAD0 = BETA0 * (6. * V0 / PI) ** (1. / 3.) / 2.
      RADIUS = RAD0

      IF (GEOMOD .EQ. 'A') CALL UPGEOA(V, RAD0, NUMCOL, DEL)
      IF (GEOMOD .EQ. 'B') THEN
        IF (THETA0 .LT. CTHETA(BETA0)) THEN
          PRINT *, 'FOR GEOMETRY MODEL B, THETA0 MUST EXCEED THE INITIAL THETA FOR MODEL A!'
          STOP
        END IF
        CALL UPGEOB('1', V, BETA0, V0, RAD0, THETA0, THETAR, NUMCOL, DEL, RADIUS, THETA)
      END IF
C.....WRITE THE INITIAL SHAPE TO THE GEOMETRY FILE (*.geo)
      WRITE (44, '(13G15.6)') TIME, DEL
      PRINT *, 'SHAPE DISTRIBUTION:'
      DO 100 J = 1,NUMCOL
        PRINT *, DEL(J) * 1000.
  100 CONTINUE
      PRINT *, ' '
      PRINT *, 'RAD0:   ', RAD0 * 1000., '   MM'

C.....ASSUME INITIAL (20C) DENSITY OF WATER IN VOLUME CALCULATIONS
      RHOH2O = 998.2
C.....USE KL AND ALPHA VALUES AT AVERAGE OF TL AND TC FOR SEMI-INFINITE SOLUTION
      PRINT *, 'TS0:', TS0, 'TL', TL
      TC = SISTC(TS0, TL)
      PRINT *, 'THE SEMI-INFINITE SOLUTION BOUNDARY TEMPERATURE IS:', TC
      KLSIS = KL((TL + TC) / 2.)
      ALFSIS = ALPHAL((TL + TC) / 2.)

C.....SET RADIATIVE PARAMETERS
      EPSF = .2261
      SIGMA = 5.67E-08
      F = .2552
      EPSILN = .84
      FRAD0 = EPSF * SIGMA * (TCOIL + 273.15) ** 4
      FRADS = F * EPSILN * SIGMA * (TCOIL + 273.15) ** 4
      HR = EPSILN * SIGMA * ((TS0 + 273.15 - 15 ) +
     +   (TAMB + 273.15)) * ((TS0 + 273.15 - 15 ) ** 2 + (TAMB + 273.15) ** 2)

C.....CALCULATE HCONV, FLUX0 (INITIAL CONSTANT FLUX)
      H = HCONV(TS0)
      HO = H + HR
      FLUX0 = -SOLVAP(TS0, HO, FRADS)
      PRINT *, 'FLUX0: ', FLUX0

C     .INITIALIZE DROPLET TIMES OF SEMI-INFINITE SOLUTION AND NFTIME FLAGS
C     .AND FIND THE NUMBER OF TRANSIENT COLUMNS INITIALLY BASED ON THE FOURIER NUMBERS
      DO 110 J = 1, NUMCOL
        TSIS(J) = (DEL(J) ** 2) / (17  * ALFSIS)
        PRINT *, 'TSIS(', J, '):', TSIS(J)
        FO = ALFSIS * DTSHRT / DEL(J) ** 2
        PRINT *, 'FO(', J, '):', FO
        IF (FO.LT.0.421) NTRANC = J
        NFTIME(J) = '0'
  110 CONTINUE
```

```
C.....INITIALLY MUST USE SHORT TIME INTERVAL
      DT = DTSHRT
      CALL WEIGHT(10, NUMNOD, DT, RAD0, W)

C.....CALCULATE THE FORCING FUNCTION AT TIME STEP 0.5
C.....AND INITIALIZE THE SOLID SURFACE TEMPERATURES
      KS = 1.297
      FLUX05 = KLSIS * (TC - TL) / (PI * ALFSIS * DT / 2.) ** .5
      DO 120 J = 1,NUMCOL
        T(J) = TC
        FRCFNC(1, J) = -(FLUX05 - FRAD0 * FRAD(DEL(J), TCOIL) + FLUX0) / KS
  120 CONTINUE
      DO 130 J = NUMCOL + 1 , NUMNOD
        T(J) = TS0
  130 CONTINUE

C.....CALL BEM1 IN ORDER TO FIND T(J) AFTER ONE TIME STEP
      CALL BEM1(FRCFNC, W, DT, RAD0, NUMNOD, NUMBET, NEGRAT, TS0, T, MEMORY, PREDU)

C.....MUST FIND CLOSE VALUES FOR THE INITIAL SSTI(J) VALUES FOR SS COLUMNS
C.....THE ONE-TERM LINEARIZED TAYLOR SERIES USED CANNOT CORRECT FOR BIG DELTA(SSTI(J))
      DO 131 J = NTRANC + 1, NUMCOL
        SSTI(J) = TL
  132   CONTINUE
          SSTI(J) = SSTI(J) + 0.1
          IF (LVFLUX(SSTI(J), PSAT, H, DEL(J), TCOIL, FRAD0) - (T(J)-SSTI(J))/DEL(J) .GE. 0.0) GOTO 133
        GOTO 132
  133   CONTINUE
        PRINT *, 'INITIAL SS SSTI(', J, '):', SSTI(J)
  131 CONTINUE

C.....TIME STEP LOOP
  140 CONTINUE
C.......IF TIME >= TSHORT THEN WANT TO USE LARGER TIME STEP
        IF (((TIME + 0.0001) .GE. TSHORT) .AND. (TSFLAG .EQ. '0')) THEN
          DT = DTLONG
          TSFLAG = '1'
          CALL WEIGHT(2, NUMNOD, DT, RAD0, W)
          CALL RECONF(FRCFNC)
        END IF

C.......STEP TIME (TO TIME CORRESPONDING TO END OF THIS ITERATION)
        TIME = TIME + DT

C.......RESET CUMULATIVE CHANGE IN VOLUME
        DV = 0.

C.......USE TRIDIAGONAL GAUSSIAN ELIMINATION FOR TRANSIENT COLUMNS
C.......ONLY IF THE HEAT WAVE HAS REACHED THE LIQUID-VAPOR INTERFACE
        DO 150 J = 1,NTRANC
C.........DEL(J) = 0 WAS ALREADY TAKEN CARE OF IN PREDICTOR LOOP
          IF (TIME .GT. TSIS(J)) THEN
            TRNBGN = '1'

            IF (NFTIME(J) EQ.'0') CALL INTCOL(J, N, TL, T(J), AOLD)
            PRINT *, 'INTCOL T(', J, ')  ', T(J)
            NFTIME(J) = '1'

            CALL LINBC(J, N, PSAT, HVFG, H, TCOIL, FRAD0, DEL, AOLD, DZ, A)

            CALL GAUSEL(J, N, DEL, DT, T, A, AOLD, DZ)

C...........SAVE TEMPERATURES IN AOLD FOR NEXT ITERATION
            DO 160 I = 1,N
              AOLD(I, 4, J) = A(I, 4, J)
```

```
  160       CONTINUE

C.....PRINT OUT TEMPERATURE COLUMN -- TRANSIENT
          DO 170 I = N,1,-1
            PRINT *, A(I, 4, J)
  170       CONTINUE
          PRINT *, ' '

C.....CALCULATE SOLID SURFACE FLUXES
          TI = (A(N, 4, J) + A(N - 1, 4, J)) / 2.
          IF (TI .GT. 99.9) THEN
            NFLUX = (KL(100.) * (A(N - 1, 4, J) - A(N, 4, J)) / DZ - H * (100. - TAMB) +
     +         FRAD0 * (1. - FRAD(DEL(J), TCOIL))) / (HFG(100.) * 18.0152)
           ELSE
            NFLUX = NH2O(TI, PSAT, H)
          END IF
          FLUX(J) = KL(T(J)) * (A(1, 4, J) - A(2, 4, J)) / DZ - FRAD0 * FRAD(DEL(J), TCOIL)
          PRINT *, 'BEM FLUX:', FLUX(J)

          DVCOL = 18.0152 * DT * NFLUX * PI * RAD0 ** 2 * (2. * J - 1.) / (NUMCOL ** 2 * RHOH2O)
          DV = DV + DVCOL
          PRINT *, 'TRANSIENT COLUMN DVCOL:', DVCOL

        ELSE
C.....USE THE SEMI-INFINITE SOLUTION FLUX
          PRINT *, 'HEAT WAVE STILL TRAVELLING IN COLUMN:    ', J
          FLUX(J) = KLSIS * (TC - TL) / (PI * ALFSIS * TIME) ** .5 - FRAD0 * FRAD(DEL(J), TCOIL)
          PRINT *, 'SISFLUX(', J, '):   ', FLUX(J)
        END IF
  150   CONTINUE
        PRINT *, ' '

C.....UPDATE INTERFACIAL TEMPERATURES FOR STEADY-STATE COLUMNS
        DO 180 J = NTRANC + 1 , NUMCOL

C.....IF SOLID IS DRY THEN USE SOLID-VAPOR BOUNDARY CONDITION
          IF (DEL(J) .EQ. 0.) THEN
            FLUX(J) = SOLVAP(T(J), HO, FRADS)
            GOTO 180
          END IF

          IF (SSTI(J) .GE. 100.) SSTI(J) = 99.9
          SSFI = LVFLUX(SSTI(J), PSAT, H, DEL(J), TCOIL, FRAD0)
          SSDFDT = DFDT(SSTI(J), PSAT, HVFG, H)
          SSA = SSDFDT
          SSB = SSFI - SSTI(J) * SSDFDT
          EXPR = -(SSA * T(J) + SSB) / (SSA * DEL(J) + 1.)
               SSTOLD = SSTI(J)
               PRINT *, 'OLD SSTI(', J, ') ', SSTI(J)
          SSTI(J) = DEL(J) * EXPR + T(J)
          IF (SSTI(J).GT.99.9) THEN
            SSTI(J) = 99.9
            KAVG = KL((T(J) + 100.) / 2 )
            QC = KAVG * (T(J) - 100.) / DEL(J)
            FLUX(J) = QC - FRAD0 * FRAD(DEL(J), TCOIL)
            NFLUX = (QC - H * (100  - TAMB) + FRAD0 * (1  - FRAD(DEL(J), TCOIL))) / (HFG(100.) * 18.0152)
            PRINT *, 'SSTI HAS REACHED 100'
            PRINT *, KAVG, T(J), DEL(J), QC, FLUX(J)
           ELSE
            PRINT *, 'SSTI(', J, ') ', SSTI(J), 'DELTSSTI ', SSTI(J) - SSTOLD
            KAVG = KL((T(J) + SSTI(J)   2 )
C           ...USE KL VALUE AT AVERAGE TEMPERATURE
            FLUX(J) = -KAVG * EXPR - FRAD0 * FRAD(DEL(J), TCOIL)
            NFLUX = NH2O(SSTI(J), PSAT, H)
          END IF
          PRINT *, 'SSFLUX:', FLUX(J)
```

```
          DVCOL = 18.0152 * DT * NFLUX * PI * RAD0 ** 2 * (2. * J - 1.) / (NUMCOL ** 2 * RHOH2O)
          DV = DV + DVCOL
          PRINT *, 'STEADY-STATE COLUMN DVCOL:', DVCOL

  180   CONTINUE

        PRINT *, 'THE LAST TRANSIENT COLUMN IS COLUMN:   ', NTRANC

C.......CHECK FLUX RATIO OF OUTERMOST TRANSIENT COLUMN TO SEE IF IT SHOULD BE SS
        IF ((NTRANC .GT. 0) .AND. (TRNBGN .EQ. '1')) THEN
          J = NTRANC
          FLUXR = (A(N, 4, J) - A(N - 1, 4, J)) / (A(2, 4, J) - A(1, 4, J))
          PRINT *, 'FLUXR(', J, '):', FLUXR

          IF (FLUXR .GT. PERC / 100.) THEN
            SSTI(J) = (A(N, 4, J) + A(N - 1, 4, J)) / 2.
C.............TRANSFER TI
            NTRANC = NTRANC - 1
            IF (NTRANC .EQ. 0) SSTIME = TIME
          END IF
        END IF

C.......UPDATE THE VOLUME OF THE DROPLET
        V = V - DV
        PRINT *, 'VOLUME:', V * 1E+09, '   MICROLITERS'
        IF ((V .LT. 0.) .AND. (EVFLAG .EQ. '0')) THEN
          PRINT *, 'THE COLUMN IS COMPLETELY EVAPORATED.'
          DO 190 J = 1,NUMCOL
            DEL(J) = 0.
  190     CONTINUE
          EVFLAG = '1'
          EVTIME = TIME
        END IF
        IF (EVFLAG .EQ. '0') THEN
          IF (GEOMOD .EQ. 'B') THEN
            CALL UPGEOB('0', V, BETA0, V0, RAD0, THETA0, THETAR, NUMCOL, DEL, RADIUS, THETA)
            IF ((THETA .LE. THETAR) .AND. (THFLAG .EQ. '0')) THEN
              THTIME = TIME
              THFLAG = '1'
            END IF
          END IF
C.........CALCULATE THE CURRENT BETA (BASED ON CURRENT V AND RADIUS)
          BETA = 2. * RADIUS / (6. * V / PI) ** (1./3.)
          IF (GEOMOD .EQ. 'A') THEN
            CALL UPGEOA(V, RAD0, NUMCOL, DEL)
            THETA = CTHETA(BETA)
          END IF
          PRINT *, 'SHAPE DISTRIBUTION:'
          DO 200 J = 1,NUMCOL
            PRINT *, DEL(J) * 1000.
  200     CONTINUE
          PRINT *, ' '
        END IF

C.......CALCULATE SURFACE FLUXES FOR R > RAD0
        DO 220 J = NUMCOL + 1 , NUMMOD
          FLUX(J) = SOLVAP(T(J), HO, FRADS)
  220   CONTINUE

C.......SAVE ALL SURFACE FLUXES IN OLDFLX()
        DO 230 J = 1,NUMMOD
          OLDFLX(J) = FLUX(J)
  230   CONTINUE

        CALL UPFFNC('1', FLUX, FLUX0, NUMBET, NUMMOD, FRCFNC)
```

```
C.......WRITE DATA TO OUTPUT FILES ***************************************************
       PRINT *, ' '
       PRINT *, '        J         FLUX(J)          T(J)          TI          SSTI'
       DO 201 J = 1, NUMCOL
         TICONV = (AOLD(N, 4, J) + AOLD(N - 1, 4, J)) / 2.
         PRINT *, J, FLUX(J), T(J), TICONV, SSTI(J)
  201  CONTINUE
       DO 206 L = 1, NDATTM
         IF (INT(10.*DATTIM(L)).EQ.INT(10.*TIME+0.01)) THEN
           DO 207 J = 1, NUMNOD
             IF (J .LE. 48) THEN
               R = RAD0 * (J - .5) / 12.
              ELSEIF (J .LE. 66) THEN
               R = RAD0 * (4. + ((J - 48) - .5) / 6.)
              ELSEIF (J .LE. 75) THEN
               R = RAD0 * (7. + ((J - 66) - .5) / 3.)
              ELSE
               R = RAD0 * (10. + ((J - 75) - .5))
             END IF
             WRITE (42, '(6G14.6)') TIME, R, R / RAD0, T(J), FLUX(J), FRCFNC(1,J)
  207      CONTINUE
           DO 204 J = 1, NUMCOL
             IF (TIME.LE.TSIS(J)) THEN
               TICONV = TL
              ELSEIF (J.LE.NTRANC) THEN
               TICONV = (AOLD(N, 4, J) + AOLD(N - 1, 4, J)) / 2.
              ELSE
               TICONV = SSTI(J)
             END IF
             WRITE (43, '(G14.6, I4, 4G14.6)') TIME, J, FLUX(J), FRCFNC(1,J), T(J), TICONV
  204      CONTINUE
         END IF
  206  CONTINUE

       WRITE (44, '(13G15.6)') TIME, DEL

       PRINT *, 'FLUXR: ', FLUXR
       PRINT *, 'VOLUME: ', V * 1E+09, 'TIME: ', TIME, ' S'
       PRINT *, 'DV: ', DV * 1E+09
       PRINT *, 'MEMORY: ', MEMORY
       PRINT *, 'BETA: ', BETA, 'THETA: ', THETA * (180. / PI)
C.......APPROXIMATE CHECK OF OVERALL ENERGY BALANCE
       TICAVG = 0.
       TAVG = 0.
       FAVG = 0.
       DO 202 J = 1, NUMCOL
         TAVG = TAVG + T(J) * (2. * J - 1.) / NUMCOL ** 2
         FAVG = FAVG + FLUX(J) * (2. * J - 1.) / NUMCOL ** 2
         CONDUC = CONDUC + DT * FLUX(J) * PI * RAD0 ** 2 * (2. * J - 1.) / NUMCOL ** 2
         RADIAT = RADIAT + DT * FRAD0 * (1. - FRAD(DEL(J), TCOIL)) * PI * RAD0 ** 2 * (2. * J - 1.) / NUMCOL ** 2
         IF (TIME.LE.TSIS(J)) THEN
           TICONV = TL
          ELSEIF (J.LE.NTRANC) THEN
           TICONV = (AOLD(N, 4, J) + AOLD(N - 1, 4, J)) / 2.
          ELSE
           TICONV = SSTI(J)
         END IF
         TICAVG = TICAVG + TICONV * (2. * J - 1.) / NUMCOL ** 2
         CONVEC = CONVEC + DT * H * (TICONV - TAMB) * PI * RAD0 ** 2 * (2. * J - 1.) / NUMCOL ** 2
  202  CONTINUE
       LATENT = LATENT + DV * 998.2 * HFG(TICAVG)
       MAXSEN = V * 998.2 * (440. - 104.89) * 1000.
       SENSIB = SENSIB + DV * 998.2 * (440. - 104.89) * 1000.
       PRINT *, 'CONDUC: ', CONDUC, 'RADIAT: ', RADIAT
       PRINT *, 'CONVEC: ', CONVEC, 'MAXSEN: ', MAXSEN, 'SENSIB: ', SENSIB
```

```
      PRINT *, 'TICAVG: ', TICAVG, 'TAVG: ', TAVG, 'LATENT: ', LATENT
      PRINT *, 'BALANCE: ', CONDUC + RADIAT - CONVEC - LATENT

      WRITE (41, '(10G14.6)') TIME, FLUXR, V * 1E9, DV * 1E9 / DT, MEMORY,
     +                  BETA, THETA * (180. / PI), TAVG, TICAVG, FAVG
      WRITE (45, '(7G14.6)') TIME, CONDUC, RADIAT, CONVEC, LATENT, SENSIB, MAXSEN
C.....*****************************************************************************

      CALL BEM1(FRCFNC, N, DT, RAD0, NUMNOD, NUMBET, NEGRAT, TS0, T, MEMORY, PREDU)

C.....+++++++++++++++NOW PREDICT THE NEXT FLUX VECTOR++++++++++++++++++++
      PRINT *, 'NOW GOING INTO PREDICTION LOOP!!!!!'
      DO 240 J = 1, NTRANC
C.....IF COLUMN EVAPORATED BEFORE STEADY-STATE THEN TRANSFER TO SS
        IF (DEL(J) .EQ. 0.) THEN
          NTRANC = J - 1
          GOTO 250
        END IF
        IF (TIME .GT. TSIS(J)) THEN
          CALL LINBC(J, N, PSAT, HVFG, H, TCOIL, FRAD0, DEL, AOLD, DZ, A)

          CALL GAUSEL(J, N, DEL, DT, T, A, AOLD, DZ)

C.....DO NOT SAVE TEMPERATURES IN AOLD FOR NEXT ITERATION

C.....PRINT OUT TEMPERATURE COLUMN -- TRANSIENT
          PRINT *, 'THESE TEMPERATURES ONLY USED TO PREDICT FLUX()'
          DO 260 I = N,1,-1
            PRINT *, A(I, 4, J)
 260      CONTINUE

C.....CALCULATE PREDICTED SOLID SURFACE FLUXES FOR R < RAD0
          FLUX(J) = KL(T(J)) * (A(1, 4, J) - A(2, 4, J)) / DZ - FRAD0 * FRAD(DEL(J), TCOIL)
          PRINT *, 'BEM PREDICTED FLUX:', FLUX(J)
        ELSE
C.....USE THE SEMI-INFINITE SOLUTION FLUX
          FLUX(J) = KLSIS * (TC - TL, * (PI * ALFSIS * TIME) ** -.5 - FRAD0 * FRAD(DEL(J), TCOIL)
          PRINT *, 'SISFLUX(', J, '      ', FLUX(J)
        END IF
 240  CONTINUE
      PRINT *, ' '

 250  CONTINUE
C.....CONTINUE PREDICTION OF FLUXES NOW IN THE STEADY-STATE MODE
      DO 270 J = NTRANC + 1 , NUMCOL

C     ...IF SOLID IS DRY THEN USE SOLID-VAPOR BOUNDARY CONDITION
        IF (DEL(J) .EQ. 0.) THEN
          FLUX(J) = SOLVAP(T(J), HC, FRADS)
          GOTO 270
        END IF

        IF (SSTI(J) .GE. 100 ) SSTI J) = 99 9
        SSFI = LVFLUX(SSTI(J), PSAT, H  DEL J   TCOIL  FRAD0)
        SSDFDT = DFDT(SSTI(J), PSAT, HVFG  H
        SSA = SSDFDT
        SSB = SSFI - SSTI(J) * SSDFDT
        EXPR = -(SSA * T(J) + SSB)     SSA * DEL J  + 1
C       DO NOT UPDATE THE STEADY-STATE LIQUID-VAPOR INTERFACIAL TEMPERATURE
        SSTEMP = DEL(J) * EXPR + T J
        IF (SSTEMP GT 99 9) THEN
          KAVG = KL((T J) + 100 )   2
          QC = KAVG * (T(J) - 100      DEL J
          FLUX(J) = QC - FRAD0 * FRAD DEL J   TCOIL
        ELSE
          KAVG = KL((T(J) + SSTI(J)) / 2
```

```
C..........OLD KAVG VALUE GOOD ENOUGH
            FLUX(J) = -KAVG * EXPR - FRAD0 * FRAD(DEL(J), TCOIL)
          END IF
          PRINT *, 'PREDICTED SSFLUX(', J, '):', FLUX(J)

  270   CONTINUE

C.......CALCULATE SURFACE FLUXES FOR R > RAD0
        DO 280 J = NUMCOL + 1 , NUMNOD
          FLUX(J) = SOLVAP(T(J), HO, FRADS)
  280   CONTINUE
C.....*****************************************************************

C.......AVERAGE FLUX AND OLDFLX TO GET FLUX AT CENTER OF CURRENT TIME INTERVAL
        DO 290 J = 1,NUMNOD
          FLUX(J) = (FLUX(J) + OLDFLX(J)) / 2.
  290   CONTINUE

        CALL UPFFNC('0', FLUX, FLUX0, NUMBET, NUMNOD, FRCFNC)

        CALL BEM2(FRCFNC, W, NUMBET, DT, RAD0, NUMNOD, TS0, T, PREDU)

        IF ((TIME + 0.0001) .GE. TEND) GOTO 9999

      GOTO 140

 9999 CONTINUE

C.....PRINT OUT RESULTS
      PRINT *, 'TIME UNTIL THE DROPLET EVAPORATED = ', EVTIME, 'S'
      PRINT *, 'TIME UNTIL QUASI-STEADY STATE WAS REACHED = ', SSTIME, 'S'
      IF (GEOMOD .EQ. 'B') PRINT *, 'TIME UNTIL THETA REACHED THETAR WAS = ', THTIME, 'S'

      STOP
      END



      REAL FUNCTION ALPHAL (T)
C.....ALPHAL  THERMAL DIFFUSIVITY OF WATER (M**2/S)
C.....USE FIFTH ORDER POLYNOMIAL TO APPROXIMATE ALPHA FOR WATER AS FUNCTION OF TEMPERATURE
      REAL T
      ALPHAL = .0000001 * (1.3514 + .0051737 * T - 3.3096E-05 * T ** 2 + 3.9658E-07 * T ** 3
     +                     - 3.863E-09 * T ** 4 + 1.3517E-11 * T ** 5)
      RETURN
      END


      SUBROUTINE BEM1 (FRCFNC, W, DT, RAD0, NUMNOD, NUMBET, NEGRAT, TS0, T, MEMORY, PREDU)
C.....PERFORM BOUNDARY ELEMENT INTEGRATION USING WEIGHT TENSOR AND FORCING FUNCTION
C.....U = T - TS0
C.....UMAX = MAX ADDITION TO U AT TIME T0
      INTEGER NUMNOD, NUMBET
      REAL FRCFNC(NUMBET, NUMNOD), W(NUMNOD, NUMNOD, 10), T(NUMNOD), PREDU(NUMNOD)
      REAL NEGRAT, TS0, DT, RAD0, MEMORY
      REAL U(78), UMAX, UTERMX
      INTEGER I, J, K
      CHARACTER*1 EXFLAG(78)
C.....U: ACCUMULATING TEMPERATURE DEPRESSION. UMAX  MAGNITUDE OF LARGEST UTERMX;
C.....UTERMX: TERM IN BEM INTEGRATION.
C.....EXFLAG: FLAG -- '1' = NODE READY TO EXIT RECOLLECTION TIME LOOP
      REAL WGHT
      UMAX = 1.E-10
      DO 400 J = 1,NUMNOD
        U(J) = 0.
```

```
      PREDU(J) = 0.
      EXFLAG(J) = '0'
400 CONTINUE
    DO 410 K = 1,NUMBET
      DO 420 J = 1,NUMNOD
        UTERMK = 0.
        DO 430 I = 1,NUMNOD
          UTERMK = UTERMK + WGHT(J, I, K, DT, RAD0, W) * FRCFNC(K, I)
430     CONTINUE
        U(J) = U(J) + UTERMK
        IF (K.GT.1) PREDU(J) = PREDU(J) + UTERMK
        IF (ABS(UTERMK) .GT. UMAX) UMAX = ABS(UTERMK)
        IF (ABS(UTERMK) / UMAX .LE. NEGRAT) EXFLAG(J) = '1'
420   CONTINUE
      DO 440 J = 1,NUMNOD
        IF (EXFLAG(J) .EQ. '0') GOTO 410
440   CONTINUE
      GOTO 450
410 CONTINUE
    PRINT *, 'NOT ENOUGH TIME STEPS IN THE INTEGRATION WERE USED!!!'
    STOP
450 CONTINUE
    DO 460 J = 1,NUMNOD
      T(J) = U(J) + TS0
460 CONTINUE
    MEMORY = K * DT
    PRINT *, 'THE NUMBER OF BEM TIME STEPS USED WAS:   ', K
    RETURN
    END


    SUBROUTINE BEM2 (FRCFNC, W, NUMBET, DT, RAD0, NUMNOD, TS0, T, PREDU)
C.....PERFORM BOUNDARY ELEMENT INTEGRATION USING WEIGHT TENSOR AND FORCING FUNCTION
C.....CALCULATIONS ARE SHORTENED BY USING PREDU FROM BEM1
C.....U = T - TS0
    INTEGER NUMNOD, NUMBET
    REAL FRCFNC(NUMBET, NUMNOD), W(NUMNOD, NUMNOD, 10), T(NUMNOD), PREDU(NUMNOD)
    REAL TS0, DT, RAD0
    REAL UTERMK
    INTEGER I, J, K
C.....UTERMK: TERM IN BEM INTEGRATION
    REAL WGHT
C.....ONLY NEED TO RECALCULATE THE FIRST TIME INTERVAL BACK IN TIME
    K = 1
    DO 470 J = 1,NUMNOD
      UTERMK = 0.
      DO 480 I = 1,NUMNOD
        UTERMK = UTERMK + WGHT(J, I, K, DT, RAD0, W) * FRCFNC(K, I)
480   CONTINUE
      T(J) = PREDU(J) + UTERMK + TS0
470 CONTINUE
    RETURN
    END


    REAL FUNCTION CTHETA (BETA)
C.....CALCULATE THE CONTACT ANGLE GIVEN BETA ASSUMING GEOMETRIC MODEL A (SEGMENT OF SPHERE)
    REAL BETA, GAM
C.....GAM: THICKNESS OF DROPLET AT APEX / RAD0
    GAM = (4./BETA ** 3 + (1. + 16./BETA **6) **.5) ** (1./3.)
   +                - (-4./BETA **3 + (1. + 16./BETA **6) **.5) ** (1./3.)
    CTHETA = ATAN(((1./GAM + GAM) **2 /4. - 1.) ** -.5)
    RETURN
    END
```

```
      REAL FUNCTION CPA (T)
C.....CPA  CONSTANT PRESSURE SPECIFIC HEAT FOR AIR (J/KG-K)
C.....USE SECOND ORDER POLYNOMIAL TO APPROXIMATE CP FOR AIR AS FUNCTION OF TEMPERATURE
      REAL T
      CPA = 1003.4 + .031646 * T + 3.4286E-04 * T ** 2
      RETURN
      END


      REAL FUNCTION DFDT (TI, PSAT, HVFG, H)
C.....CALCULATE DERIVATIVE OF VAPOR-LIQUID FLUX WRT TEMPERATURE
      REAL TI, PSAT(0:100), HVFG(0:100), H
      REAL TAMB, TF, XA, PATM, LE23, COEFF, XTERM
C.....TAMB: AMBIENT TEMP; TF: BOUNDARY LAYER FILM TEMP; XA: AMBIENT MOLE FRACTION OF WATER;
C.....PATM: ATMOSPHERIC PRESSURE; LE23: LEWIS NUMBER ^ (2/3);
C.....COEFF, XTERM: ALGEBRAIC EXPRESSIONS
      REAL KL
      TAMB = 25.
      TF = (TI + TAMB) / 2.
      XA = 0.
      PATM = 101325.
      LE23 = .894
      COEFF = .621924 * H * HFG(TI) / (CPA(TF) * LE23)
      XTERM = (1. - XI(TI, PSAT)) ** 2
      DFDT = (COEFF * (1 - XA) * HV(TI, HVFG) / (PATM * XTERM * (TI + 273.15)) + H) / KL(TI)
C.....CAN USE  DFDT = H / KL(TI)  AS A CHECK FOR CASE OF CONVECTION ONLY
      RETURN
      END


      REAL FUNCTION FRAD (D, T)
C.....CALCULATE FRACTION OF RADIATIVE FLUX REACHING DEPTH D IN DROPLET
C.....FUNCTION IS A 2D CURVE FIT OF RADIATION STUDY RESULTS
      REAL D, T
      REAL Z, M, B
C.....D: DEPTH BELOW LIQUID-VAPOR INTERFACE (M)
C.....Z: DEPTH (MM)
C.....T: TEMPERATURE OF RADIATIVE HEATER COILS (C)
C.....FRAD: NORMALIZED RADIATIVE FLUX (FLUX/[(EPS*F)*SIGMA*(T+273.15)**4])
      Z = D * 1000.
      IF (Z .LE. .04) THEN
        M = 322.37 - .58492 * T + 3.2645E-04 * T ** 2
        B = 0.
       ELSEIF (Z .LE. .1) THEN
        M = 1042.3 - 2.7002 * T + .0018666 * T ** 2
        B = -61.573 + .25781 * T - 3.6576E-04 * T ** 2 + 1.7732E-07 * T ** 3
       ELSEIF (Z .LE. .2) THEN
        M = 3863.7 - 16.516 * T + .024457 * T ** 2 - 1.2372E-05 * T ** 3
        B = -273. + 1.2587 * T - .0019433 * T ** 2 + 1.0096E-06 * T ** 3
       ELSEIF (Z .LE. .7) THEN
        M = 4122.1 - 18.206 * T + .027637 * T ** 2 - 1.6258E-05 * T ** 3
        B = -309.61 + 1.5276 * T - .0024716 * T ** 2 + 1.3303E-06 * T ** 3
       ELSE
        M = 5577.9 - 24.76 * T + .037703 * T ** 2 - 1.9493E-05 * T ** 3
        B = -1336. + 6.1489 * T - .0095698 * T ** 2 + 5.0226E-06 * T ** 3
      END IF
      FRAD = 1. / (M * Z + B + 1.)
      RETURN
      END


      SUBROUTINE GAUSEL (J, N, DEL, DT, T, A, AOLD, DZ)
C     USE TRIDIAGONAL GAUSSIAN ELIMINATION TO UPDATE TEMPERATURES IN DROPLET COLUMN
      PARAMETER (NUMCOL = 12)
      INTEGER J, N
      REAL DEL(NUMCOL), DT, T(NUMCOL), A(12, 4, NUMCOL), AOLD(12, 4, NUMCOL), DZ
```

```
      REAL TZ2, ALPHA, GAMMA
C.....TZ2: CRANK-NICHOLSON EXPRESSION; ALPHA: THERMAL DIFFUSIVITY OF LIQUID;
C.....GAMMA: CRANK-NICHOLSON CONSTANT
      REAL ALPHAL
      A(1, 1, J) = 0.
      A(1, 2, J) = .5
      A(1, 3, J) = .5
      A(1, 4, J) = T(J)
      TZ2 = DT / (2. * DZ ** 2)
C.....THE C-N ALPHA IS A MEASURE OF ACCURACY OF THE CRANK-NICHOLSON METHOD
C.....GIVE A CONSERVATIVE (UPPER LIMIT) VALUE
      PRINT *, 'C-N ALPHA(', J, '):   ', 2. * 1.7E-07 * TZ2
      DO 500 I = 2 , N - 1
        ALPHA = ALPHAL(AOLD(I, 4, J))
        GAMMA = ALPHA * TZ2
        A(I, 1, J) = -GAMMA
        A(I, 2, J) = 1. + 2. * GAMMA
        A(I, 3, J) = -GAMMA
        A(I, 4, J) = AOLD(I, 4, J) + GAMMA * (AOLD(I - 1, 4, J) - 2. * AOLD(I, 4, J) + AOLD(I + 1, 4, J))
  500 CONTINUE
C.....**************MEAT OF GAUSSIAN ELIMINATION*****************
      DO 510 I = 2,N
        A(I, 2, J) = A(I, 2, J) - A(I, 1, J) / A(I - 1, 2, J) * A(I - 1, 3, J)
        A(I, 4, J) = A(I, 4, J) - A(I, 1, J) / A(I - 1, 2, J) * A(I - 1, 4, J)
  510 CONTINUE
      A(N, 4, J) = A(N, 4, J) / A(N, 2, J)
      DO 520 I = N - 1 , 1 , -1
        A(I, 4, J) = (A(I, 4, J) - A(I, 3, J) * A(I + 1, 4, J)) / A(I, 2, J)
  520 CONTINUE
C.....*************************************************************
      RETURN
      END


      REAL FUNCTION HCONV (T)
C.....HCONV  EXPERIMENTAL CONVECTIVE HEAT TRANSFER COEFFICIENT (W/M**2-K)
C.....USE THIRD ORDER POLYNOMIAL FOR EXPERIMENTAL HCONV AS FUNCTION OF TS0
      REAL T
      HCONV = -42.348 + 1.3663 * T - .011498 * T ** 2 + 3.1954E-05 * T ** 3
      RETURN
      END


      REAL FUNCTION HFG (TI)
C.....CALCULATE THE LATENT HEAT OF WATER (J/KG) AS A FUNCTION OF TEMPERATURE
      REAL TI
      REAL T, A1, A2, A3, A4, A5, A6, LAMBDA
      T = (647.27 - (TI + 273.15)) / 647.27
      A1 = .72241
      A2 = 5.33402
      A3 = 8.97347
      A4 = -11.93143
      A5 = -3.31206
      A6 = 1.63257
      LAMBDA = A1 * T **(1./3.) + A2 * T ** 79 + A3 * T **1.208 + A4 * T + A5 * T **2 + A6 * T **3
      HFG = LAMBDA * 2501000.
      RETURN
      END


      REAL FUNCTION HV (TI, HVFG)
C.....FUNCTION TO GIVE HFG/VFG GIVEN TEMPERATURE
C.....HV: RATIO OF LATENT HEAT TO SPECIFIC VOLUME CHANGE OF VAPORIZATION
      REAL TI, HVFG(0:100)
      INTEGER TSAT
      REAL TVAL, HVA, HVB
```

```
      TSAT = -1
      TVAL = TI
      IF (TVAL .GT. 100.) TVAL = 100.
  550 CONTINUE
        TSAT = TSAT + 1
      IF (TSAT .LT. TVAL - 1.) GOTO 550
      HVA = HVFG(TSAT) * 1000.
      HVB = HVFG(TSAT + 1) * 1000.
C.....LINEARLY INTERPOLATE HV
      HV = HVA + (HVB - HVA) * (TI - TSAT)
      RETURN
      END


      SUBROUTINE INTCOL (J, N, TL, TC, AOLD)
C.....CALCULATE TSIS(J), THE TIME UNTIL HEAT WAVE IN COLUMN J REACHES SURFACE AND
C.....INITIALIZE TRIDIAGONAL MATRIX USING SIS SOLUTION TEMPERATURE PROFILE
      PARAMETER (NUMCOL = 12)
      INTEGER J, N
      REAL TL, TC, AOLD(12, 4, NUMCOL)
      REAL ERF
C.....ERF: ERROR FUNCTION (FROM LINKED LIBRARY)
      DO 600 I = 2 , N - 1
        AOLD(I, 4, J) = (TL - TC) * ERF((I - 1.5) * 17. **.5 / (2. * (N - 2.))) + TC
  600 CONTINUE
C.....ADJUST T1 AND TN IN ORDER TO GET CORRECT BOUNDARY TEMPERATURES
      AOLD(1, 4, J) = 2. * TC - AOLD(2, 4, J)
      AOLD(N, 4, J) = 2. * TL - AOLD(N - 1, 4, J)
      RETURN
      END


      REAL FUNCTION KL (T)
C.....KL  THERMAL CONDUCTIVITY OF WATER (W/M-K)
C.....USE FIFTH ORDER POLYNOMIAL TO APPROXIMATE K FOR WATER AS FUNCTION OF TEMPERATURE
      REAL T
      KL = .56971 + .001754 * T  - 4.0332E-06 * T  **2 - 4.3732E-08 * T  **3
     +                           + 2.202E-10 * T  **4 - 2.9455E-13 * T  **5
      RETURN
      END


      SUBROUTINE LINBC (J, N, PSAT, HVFG, H, TCOIL, FRAD0, DEL, AOLD, DZ, A)
C.....LINEARIZE THE VAPOR BOUNDARY CONDITION USING FIRST TERM OF TAYLOR SERIES
      PARAMETER (NUMCOL = 12)
      INTEGER J, N
      REAL PSAT(0:100), HVFG(0:100), H, TCOIL, FRAD0, DEL(NUMCOL)
      REAL AOLD(12, 4, NUMCOL), DZ, A(12, 4, NUMCOL)
      REAL TI0, FI0, DFDT0, AA, BB
C.....TI0, FI0, DFDT0, AA, BB: PARAMETERS FOR TAYLOR LINEARIZATION OF LIQUID-VAPOR BC
      REAL DFDT, LVFLUX
      DZ = DEL(J) / (N - 2.)
          PRINT *, 'DZ(', J, '):', DZ * 1000., 'MM'
      TI0 = (AOLD(N, 4, J) + AOLD(N - 1, 4, J)) / 2.
          PRINT *, 'TI0:', TI0, 'T(', J, '):', (AOLD(1, 4, J) + AOLD(2, 4, J)) / 2.
      IF (TI0 .GT. 99.9) TI0 = 99.9
      FI0 = LVFLUX(TI0, PSAT, H, DEL(J), TCOIL, FRAD0)
      DFDT0 = DFDT(TI0, PSAT, HVFG, H)
      AA = DZ * DFDT0
      BB = DZ * (FI0 - TI0 * DFDT0)
      A(N, 1, J) = 1. - AA / 2.
      A(N, 2, J) = -1. - AA / 2.
      A(N, 3, J) = 0.
      A(N, 4, J) = BB
      RETURN
      END
```

```
      REAL FUNCTION LVFLUX (TI, PSAT, H, D, TCOIL, FRAD0)
C.....FUNCTION TO CALCULATE FLUX AT LIQUID-VAPOR INTERFACE
      REAL TI, PSAT(0:100), H, D, TCOIL, FRAD0
      REAL TAMB, TF, XA, LE23, COEFF, XTERM, CONV, RAD
C.....TAMB: AMBIENT TEMP; TF: BOUNDARY LAYER FILM TEMP; XA: AMBIENT MOLE FRACTION OF WATER;
C.....LE23: LEWIS NUMBER ^ (2/3); COEFF, XTERM: ALGEBRAIC EXPRESSIONS;
C.....CONV: CONVECTIVE HEAT TRANSFER; RAD: RADIATIVE HEAT TRANSFER
      REAL HFG, CPA, XI, FRAD, KL
      TAMB = 25.
      TF = (TI + TAMB) / 2.
      XA = 0.
      LE23 = .894
      COEFF = .621924 * H * HFG(TI) / (CPA(TF) * LE23)
      XTERM = (XI(TI, PSAT) - XA) / (1 - XI(TI, PSAT))
      CONV = H * (TI - TAMB)
      RAD = FRAD0 * (1. - FRAD(D, TCOIL))
      LVFLUX = (COEFF * XTERM * CONV - RAD) / KL(TI)
      RETURN
      END


      REAL FUNCTION NH2O (TI, PSAT, H)
C.....CALCULATE MOLAR FLUX AT LIQUID-VAPOR INTERFACE FOR TI < 100
      REAL TI, PSAT(0:100), H
      REAL TAMB, TF, XA, LE23, MA, COEFF, XTERM
C.....TAMB: AMBIENT TEMP; TF: BOUNDARY LAYER FILM TEMP; XA: AMBIENT MOLE FRACTION OF WATER;
C.....LE23: LEWIS NUMBER ^ (2/3); MA: MOLECULAR MASS OF AIR
C.....COEFF, XTERM: ALGEBRAIC EXPRESSIONS
      REAL CPA, XI
      IF (TI .GE. 100.) THEN
        PRINT *, 'NH2O CALCULATION TRIED FOR TI >= 100. !!'
        STOP
      END IF
      TAMB = 25.
      TF = (TI + TAMB) / 2.
      XA = 0.
      LE23 = .894
      MA = 28.9669
      COEFF = H / (MA * CPA(TF) * LE23)
      XTERM = (XI(TI, PSAT) - XA) / (1 - XI(TI, PSAT))
      NH2O = COEFF * XTERM
      RETURN
      END


      SUBROUTINE RECONF (FRCFNC)
C     FIX THE FORCING FUNCTION FOR THE SWITCH TO DTLONG
C     WARNING: GOOD ONLY FOR DTSHORT = 0.1, DTLONG = 1.0, AND TSHORT = 4. !!!!!!!
      REAL FRCFNC(100, 78)
      INTEGER J, K
      DO 610 J = 1,78
        DO 620 K = 1,4
          FRCFNC(K, J) = (FRCFNC(10*K-5, J) + FRCFNC(10*K-4, J)) / 2
  620   CONTINUE
        DO 630 K = 5,100
          FRCFNC(K, J) = 0
  630   CONTINUE
  610 CONTINUE
      RETURN
      END


      REAL FUNCTION SISTC (TS0, TL
C     CALCULATE THE SEMI-INFINITE SOLUTION CONTACT TEMPERATURE OF DROPLET AND SOLID
```

```
C.....ASSUMING INITIAL TEMPERATURE OF SOLID IS CONSTANT  IT IS ACTUALLY LINEAR/
      REAL TS0, TL
      REAL RHOS, CPS, KS, RHOL, CPL, RADS, RADL, NUM, DEN
C.....RHOS: DENSITY OF SOLID; CPS: SPECIFIC HEAT OF SOLID; KS: CONDUCTIVITY OF SOLID;
C.....RHOL: DENSITY OF LIQUID; CPL: SPECIFIC HEAT OF LIQUID;
C.....RADS, RADL, NUM, DEN: ALGEBRAIC EXPRESSIONS
      REAL KL
      RHOS = 2520.
      CPS = 888.9
      KS = 1.297
      RHOL = 998.2
      CPL = 4179.
      RADS = (RHOS * CPS * KS) ** .5
      RADL = (RHOL * CPL * KL(TL)) ** .5
      NUM = TS0 * RADS + TL * RADL
      DEN = RADS + RADL
      SISTC = NUM / DEN
      RETURN
      END


      REAL FUNCTION SOLVAP (T, HO, FRADS)
C.....CALCULATE THE FLUX FOR DRY SURFACE AT TEMPERATURE T
C.....SOLVAP IS POSITIVE OUT OF THE SURFACE, SOLVAP < 0
      REAL T, HO, FRADS
      REAL TAMB
C.....TAMB: AMBIENT TEMPERATURE
      TAMB = 25.
      SOLVAP = HO * (T - TAMB) - FRADS
      RETURN
      END


      SUBROUTINE UPFFNC (RLLDWN, FLUX, FLUX0, NUMBET, NUMNOD, FRCFNC)
C.....UPDATE THE ARRAY OF TEMPERATURE GRADIENT FORCING FUNCTIONS
C.....THROW OUT THE OLDEST IF RLLDWN = '1'
      CHARACTER*1 RLLDWN
      INTEGER NUMBET, NUMNOD
      REAL FLUX(78), FLUX0, FRCFNC(100, 78)
      REAL KS
      INTEGER I, J
C.....KS: CONDUCTIVITY OF SOLID
      KS = 1.297
      IF (RLLDWN .EQ. '1') THEN
        DO 650 I = NUMBET,2,-1
          DO 660 J = 1,NUMNOD
            FRCFNC(I, J) = FRCFNC(I - 1, J)
  660     CONTINUE
  650   CONTINUE
      END IF
      DO 670 J = 1,NUMNOD
        FRCFNC(1, J) = -(FLUX(J) + FLUX0) / KS
  670 CONTINUE
      RETURN
      END


      SUBROUTINE UPGEOA (V, RAD0, NUMCOL, DEL)
C.....UPDATE THE HEIGHT OF THE DROPLET USING MODEL 'A'
      PARAMETER (PI = 3.14159265358979)
      REAL V, RAD0, DEL(12), BETA
      INTEGER NUMCOL
      REAL GAM, R
C.....GAM: THICKNESS OF DROPLET AT APEX / RAD0, R: NONDIMENSIONAL RADIAL POSITION
      PRINT *, ' '
      BETA = 2. * RAD0 / (6. * V / PI) ** (1./3.)
```

```
      PRINT *, 'BETA:    ', BETA
      GAM = (4./ BETA**3 + (1. + 16./ BETA**6) ** 5) **(1./3.)
     +            - (-4./ BETA**3 + (1. + 16./ BETA**6) **.5) **(1./3.)
      DO 700 J = 1,NUMCOL
        R = (2. * J - 1.) / (2. * NUMCOL)
        DEL(J) = ((1./GAM + GAM) **2 /4. - R **2) **.5 - (1./GAM - GAM)/2.
C.......MAKE DEL() DIMENSIONAL
        DEL(J) = RAD0 * DEL(J)
  700 CONTINUE
      RETURN
      END


      SUBROUTINE UPGEOB (L, V, B0, V0, R0, T0, TR, NUMCOL, DEL, R, THETA)
C.....UPDATE THE HEIGHT OF THE DROPLET USING MODLE 'B'
C.....ALSO CALCULATE THE CURRENT CONTACT ANGLE THETA IN RADIANS
C.....THIS SUBROUTINE WRITTEN BY SUSAN TINKER
C.....AND TRANSLATED FROM QUICKBASIC TO FORTRAN BY GLENN WHITE
C.....DEFINE COMMON BLOCK TO ALLOW SUBROUTINE UPGEOB TO KEEP VALUES OF CERTAIN VARIABLES STATIC
      REAL VR, VL, S0, Z0, X0, A0, L1, L2, SR, AR, ZR, XR, RL, ZL, SL
      COMMON /GEOBBL/ VR, VL, S0, Z0, X0, A0, L1, L2, SR, AR, ZR, XR, RL, ZL, SL
      PARAMETER (PI = 3.14159265358979)
      REAL V, B0, V0, R0, T0, TR, DEL(12), R, THETA
      INTEGER NUMCOL
      CHARACTER*1 L
C.....*******SECTION I - DETERMINATION OF INITIAL DROPLET PARAMETERS*********
C.....DEFINITION OF VARIABLES IN SECTION I
C.....T0: INITIAL CONTACT ANGLE; INPUT BY USER
C.....V0: INITIAL DROPLET VOLUME; INPUT BY USER
C.....B0: D/D, WHERE D= DROPLET DIAMETER BEFORE IMPACT
C.....D= MAX DROPLET DIAMETER AFTER IMPACT, INPUT BY USER
C.....TR: RECEDING ANGLE; INPUT BY USER
C.....V: ANY DROP VOLUME BETWEEN 0 AND V0, INPUT BY USER
C.....R0: MAX DROPLET RADIUS AFTER IMPACT
C.....Z0: DISTANCE FROM CENTER OF INITIAL ARC TO R-AXIS
C.....X0: RADIUS OF DISK PART OF DROPLET
C.....S0: HEIGHT OF INITIAL DROPLET
C.....D0: DISTANCE ON R-AXIS FROM X0 TO RP0
C.....A0: DISTANCE ON R-AXIS FROM RP0 TO R
C.....RP0: R-COORDINATE OF P0
      REAL A, C, CL, D0, DR, D, E, F, GAM, G, RP0, RPR, RP, R1, R2, RV, ROV
      REAL SLP, S, T, U1, UR, U2, VA1, VB1, VC1, VD1, V1, VAR, VBR, VCR, VDR
      REAL VA2, VB2, VC2, VD2, V2, X, XC, Y1, YR, Y2, ZC
      INTEGER J
      IF (L .EQ. '0') GOTO 1500
      C = V0 / (PI * R0 ** 2)
      E = (3 * V0) / (PI * R0 ** 2)
 1010 S0 = (C + E) / 2.
      Z0 = S0 * ((1. / TAN(T0)) + SQRT(1. + (1. / TAN(T0)) ** 2)) * TAN(PI / 2. - T0)
      Y1 = (Z0 + S0)
      X0 = R0 - S0 * ((1. / TAN(T0)) + SQRT(1. + (1. / TAN(T0)) ** 2))
      U1 = (R0 - X0) / Y1
      VA1 = (PI * X0 ** 2 * S0)
      VB1 = 2. * PI * ((Y1 ** 3 / 3.) - (Y1 ** 3 / 3.) * (COS(ATAN(U1 / SQRT(1. - U1 ** 2)))) ** 3)
      VC1 = PI * Y1 ** 2 * X0 * (ATAN(U1 / SQRT(1. - U1 ** 2))
     +                + (1. / 2.) * SIN(2. * (ATAN(U1 / SQRT(1. - U1 ** 2)))))
      VD1 = PI * Z0 * (R0 ** 2 - X0 ** 2)
      V1 = VA1 + VB1 + VC1 - VD1
      IF ((V1 - V0) .GT. 1E-12) THEN
        E = S0
        GOTO 1010
       ELSEIF ((V0 - V1) .GT. 1E-12) THEN
        C = S0
        GOTO 1010
       ELSE
        GOTO 1020
```

```
      END IF
 1020 A0 = S0 / TAN(T0)
      D0 = S0 * SQRT(1. + (1. / TAN(T0)) ** 2)
      RP0 = R0 - A0
C.....*****END SECTION I**********************************************************
C.....*****SECTION II - CALCULATE VL AND VR*********************************
C.....TR: RECEDING ANGLE; INPUT BY USER
C.....SR: HEIGHT OF DROPLET AT RECEDING SHAPE
C.....AR: DISTANCE ON R-AXIS FROM PR TO R
C.....RPR: R-COORDINATE OF POINT PR
C.....VR: VOLUME OF DROPLET AT RECEDING ANGLE
C.....VL: VOLUME OF DROPLET AT LENS SHAPE
C.....RL: RADIUS AT LENS SHAPE IF NOT R0
C.....SL: HEIGHT OF LENS
      L1 = R0 - S0 / TAN(TR)
      L2 = SQRT((S0 / TAN(TR)) ** 2 + S0 ** 2)
      IF (L2 .GE. L1) THEN
        SR = (R0 / TAN(TR)) * ((SQRT(1. + (TAN(TR)) ** 2)) - 1.)
        AR = SR / TAN(TR)
        ZR = R0 / TAN(TR)
        RPR = R0 - AR
        VR = (1. / 6.) * PI * SR * (3. * R0 ** 2 + SR ** 2)
        VL = VR
        SL = SR
        ZL = ZR
       ELSE
        SR = S0
        AR = SR / TAN(TR)
        RPR = R0 - AR
        DR = SR * SQRT(1. + (1. / TAN(TR)) ** 2)
        ZR = (DR + AR) * TAN(PI / 2. - TR)
        XR = R0 - DR - AR
        YR = ZR + SR
        UR = (R0 - XR) / YR
        VAR = PI * XR ** 2 * SR
        VBR = 2. * PI * ((YR ** 3 / 3.) - (YR ** 3 / 3.) * (COS(ATAN(UR / SQRT(1. - UR ** 2)))) ** 3)
        VCR = PI * YR ** 2 * XR * (ATAN(UR / SQRT(1. - UR ** 2))
     *                + (1. / 2.) * SIN(2. * (ATAN(UR / SQRT(1. - UR ** 2)))))
        VDR = PI * ZR * (R0 ** 2 - XR ** 2)
        VR = VAR + VBR + VCR - VDR
        CL = R0 - S0 * ((1. / TAN(TR)) + SQRT(1. + (1. / TAN(TR)) ** 2))
        RL = R0 - CL
        ZL = RL / TAN(TR)
        SL = S0
        VL = (PI / 6.) * SR * (3. * RL ** 2 + SR ** 2)
       END IF
C.....******END SECTION II*************************************************************
C.....****SECTION III - CONFIGURATION FOR ANY VOLUME BETWEEN 0 AND V0*****************
 1500 IF (V .EQ. V0) THEN
C.....****INITIAL CONFIGURATION*********************************************************
        DO 1001 J = 1, NUMCOL
          X = R0 * (2. * J - 1.) / (2. * NUMCOL)
          IF (X .LE. X0) THEN
            DEL(J) = S0
           ELSE
            DEL(J) = SQRT((Z0 + S0) ** 2 - (X - X0) ** 2) - Z0
          END IF
 1001   CONTINUE
        R = R0
        GAM = S0 / (R0 - X0)
        GOTO 1000
       END IF
       IF (V .GT. VL) THEN
         IF (V .GT. VR) THEN
           IF (L2 .GE. L1) THEN
             SLP = (S0 - SR) / (AR - A0)
```

```
          F = TR
          G = T0
1030      T = (F + G) / 2.
          A = (A0 + (1. / SLP) * S0) / (1. + (1. / SLP) * TAN(T))
          S = A * TAN(T)
          D = SQRT(S ** 2 + A ** 2)
          XC = R0 - D - A
          RP = R0 - A
          ZC = (D + A) / TAN(T)
          Y2 = (ZC + S)
          U2 = (R0 - XC) / Y2
          VA2 = (PI * XC ** 2 * S)
          VB2 = 2. * PI * (Y2 ** 3 / 3.) * (1. - (COS(ATAN(U2 / SQRT(1. - U2 ** 2)))) ** 3)
          VC2 = PI * Y2 ** 2 * XC * (ATAN(U2 / SQRT(1. - U2 ** 2))
     +                    + (1. / 2.) * SIN(2. * (ATAN(U2 / SQRT(1. - U2 ** 2)))))
          VD2 = PI * ZC * (R0 ** 2 - XC ** 2)
          V2 = VA2 + VB2 + VC2 - VD2
          IF ((V2 - V) .GT. 1E-12) THEN
            G = T
            GOTO 1030
           ELSEIF ((V - V2) .GT. 1E-12) THEN
            F = T
            GOTO 1030
           ELSE
            GOTO 1050
           END IF
         ELSEIF (L1 .GT. L2) THEN
          F = TR
          G = T0
1040      T = (F + G) / 2.
          S = S0
          A = S / TAN(T)
          D = SQRT(S ** 2 + A ** 2)
          XC = R0 - D - A
          RP = R0 - A
          ZC = (D + A) / TAN(T)
          Y2 = ZC + S
          U2 = (R0 - XC) / Y2
          VA2 = PI * XC ** 2 * S
          VB2 = 2. * PI * (Y2 ** 3 / 3.) * (1. - (COS(ATAN(U2 / SQRT(1. - U2 ** 2)))) ** 3)
          VC2 = Y2 ** 2 * PI * XC * (ATAN(U2 / SQRT(1. - U2 ** 2))
     +                    + (1. / 2.) * SIN(2. * (ATAN(U2 / SQRT(1. - U2 ** 2)))))
          VD2 = PI * ZC * (R0 ** 2 - XC ** 2)
          V2 = VA2 + VB2 + VC2 - VD2
          IF ((V2 - V) .GT. 1E-12) THEN
            G = T
            GOTO 1040
           ELSEIF ((V - V2) .GT. 1E-12) THEN
            F = T
            GOTO 1040
           ELSE
            GOTO 1050
           END IF
         ELSE
          CONTINUE
         END IF
C.....*******CONFIGURATION FOR VOLUME V > VR*********************************************
 1050    DO 1002 J = 1, NUMCOL
          X = R0 * (2. * J - 1.) / (2. * NUMCOL)
          IF (X .LE. XC) THEN
            DEL(J) = S
           ELSE
            DEL(J) = SQRT((ZC + S) ** 2 - (X - XC) ** 2) - ZC
           END IF
 1002    CONTINUE
         R = R0
```

```
          GAM = S / (R0 - XC)
         ELSEIF (V .EQ. VR) THEN
          DO 1003 J = 1, NUMCOL
            X = R0 * (2. * J - 1.) / (2. * NUMCOL)
            IF (X .LE. XR) THEN
              DEL(J) = SR
             ELSE
              DEL(J) = SQRT((ZR + SR) ** 2 - (X - XR) ** 2) - ZR
            END IF
 1003      CONTINUE
          R = R0
          GAM = SR / (R0 - XR)
         ELSE
          R1 = RL
          R2 = R0
 1100     RV = (R1 + R2) / 2.
          S = S0
          A = S / TAN(TR)
          RP = RV - A
          D = S * SQRT(1. + (1. / TAN(TR)) ** 2)
          ZC = (D + A) * TAN(PI / 2. - TR)
          XC = RV - D - A
          Y2 = ZC + S
          U2 = (RV - XC) / Y2
          VA2 = PI * XC ** 2 * S
          VB2 = 2. * PI * ((Y2 ** 3 / 3.) - (Y2 ** 3 / 3.) * (COS(ATAN(U2 / SQRT(1. - U2 ** 2)))) ** 3)
          VC2 = PI * Y2 ** 2 * XC * (ATAN(U2 / SQRT(1. - U2 ** 2))
     +                    + (1. / 2.) * SIN(2. * (ATAN(U2 / SQRT(1. - U2 ** 2)))))
          VD2 = PI * ZC * (RV ** 2 - XC ** 2)
          V2 = VA2 + VB2 + VC2 - VD2
          IF ((V2 - V) .GT. 1E-12) THEN
            R2 = RV
            GOTO 1100
           ELSEIF ((V - V2) .GT. 1E-12) THEN
            R1 = RV
            GOTO 1100
           ELSE
            GOTO 1200
          END IF
C.....*******CONFIGURATION FOR V < VR*************************************************
 1200     DO 1004 J = 1, NUMCOL
            X = R0 * (2. * J - 1.) / (2. * NUMCOL)
            IF (X .LE. XC) THEN
              DEL(J) = S
             ELSEIF (X .LE. RV) THEN
              DEL(J) = SQRT((ZC + S) ** 2 - (X - XC) ** 2) - ZC
             ELSE
              DEL(J) = 0.
            END IF
 1004      CONTINUE
          R = RV
          GAM = S / (RV - XC)
        END IF
       ELSEIF (V .EQ. VL) THEN
C.....********CONFIGURATION FOR V = VL*****************************************
        DO 1005 J = 1, NUMCOL
          X = R0 * (2. * J - 1.) / (2. * NUMCOL)
          IF (X .LE. RL) THEN
            DEL(J) = SQRT((ZL + SL) ** 2 - X ** 2) - ZL
           ELSE
            DEL(J) = 0.
          END IF
 1005    CONTINUE
        R = RL
        GAM = SL / RL
       ELSE
```

```
C.....********CONFIGURATION FOR V < VL*****************************************
      ROV = (V / ((PI / 6.) * (1. - COS(TR)) * (3. * (SIN(TR)) ** 2 + (1. - COS(TR)) ** 2))) ** (1.   3.)
      RV = ROV * SIN(TR)
      S = ROV * (1. - COS(TR))
      ZC = ROV - S
      DO 1006 J = 1, NUMCOL
        X = R0 * (2. * J - 1.) / (2. * NUMCOL)
        IF (X .LE. RV) THEN
          DEL(J) = SQRT((ZC + S) ** 2 - X ** 2) - ZC
         ELSE
          DEL(J) = 0.
        END IF
 1006   CONTINUE
      R = RV
      GAM = S / RV
    END IF
C.....********END SECTION III*****************************************************
 1000 CONTINUE
C.....CALCULATE THETA FROM GAM, THE RATIO OF APEX TO DELTA(R) OF CURVED SURFACE
      THETA = ATAN(((1./GAM + GAM) **2 /4. - 1.) ** -.5)
      RETURN
      END


      SUBROUTINE WEIGHT (NUMWTI, NUMNOD, DT, RAD0, W)
C.....CALCULATE THE WEIGHT MATRIX W CAREFULLY FOR THE FIRST NUMWTI TIME STEPS
C.....WARNING: GOOD ONLY FOR DTSHORT = 0.1 AND DTLONG = 1.0 !!!!!!!
      PARAMETER (PI = 3.14159265358979)
      INTEGER NUMWTI, NUMNOD
      REAL DT, RAD0, W(78, 78, 10)
C.....NUMWTI: NUMBER OF TIME STEPS TO BE PRECALCULATED
      REAL ALPHA, DT0, T0, CONST1, DR0, R0, R, ARG1, ARG2, ARG3, SUM
      INTEGER I, J, K, L, LMAX
C.....ALPHA: THERMAL DIFFUSIVITY OF SOLID; DT0: RECOLLECTION TIME INTERVAL;
C.....T0: RECOLLECTION TIME; CONST1: ALGEBRAIC EXPRESSION FROM GREEN'S FUNCTION;
C.....DR0: DUMMY RADIAL POSITION INTERVAL, R0: DUMMY RADIAL POSITION;
C.....R: RADIAL POSITION; ARG1, ARG2, ARG3  GREEN'S FUNCTION ARGUMENTS;
C.....SUM: INTEGRATION SUMMATION; L: INTEGRATION INDEX, LMAX: NUMBER OF STEPS PER TIME INTEGRATION
      REAL ERF, BESI0E
C.....ERF: ERROR FUNCTION; BESI0E: EXPONENTIAL BESSEL FUNCTION I0(ARG) X EXP(-ARG)
C.....(FROM LINKED LIBRARY)
      IF (DT.EQ.0.1) LMAX = 25
      IF (DT.EQ.1.0) LMAX = 250
      ALPHA = 5.79E-07
      DT0 = DT
      CONST1 = (4 * PI * ALPHA) ** - 5
      DO 750 K = 1,NUMWTI
        DO 760 I = 1,78
          DO 770 J = 1,78
            SUM = 0.
            IF (I  LE  48) THEN
              DR0 = RAD0 / 12
              R0 = RAD0 * (I -  5) / 12
             ELSEIF (I  LE  66) THEN
              DR0 = RAD0 / 6
              R0 = RAD0 * (4  + (I - 48  -  5    6
             ELSEIF (I  LE  75) THEN
              DR0 = RAD0 / 3
              R0 = RAD0 * (7  + (I - 66  -  5    3 .
             ELSE
              DR0 = RAD0
              R0 = RAD0 * (10  + (I   75  -  5
            END IF
            IF (J  LE  48) THEN
              R = RAD0 * (J -  5)    12
             ELSEIF (J  LE  66) THEN
```

```
            R = RAD0 * (4. + ((J - 48) - .5) / 6.)
           ELSEIF (J .LE. 75) THEN
            R = RAD0 * (7. + ((J - 66) - .5) / 3.)
           ELSE
            R = RAD0 * (10. + ((J - 75) - .5))
           END IF
           IF (I .EQ. J) THEN
             DO 780 L = 1,LMAX
             T0 = (K - 1 + L / (1. * LMAX)) * DT0
             ARG1 = R * R0 / (2. * ALPHA * T0)
             ARG2 = (DR0 / 2.) / (4. * ALPHA * T0) ** .5
780          SUM = SUM + T0 ** -1 * BESI0E(ARG1) * ERF(ARG2)
             W(J, I, K) = R * SUM * DT0 / (1. * LMAX)
                 PRINT *, J, I, K, W(J, I, K)
            ELSE
             DO 790 L = 1,LMAX
             T0 = (K - 1 + L / (1. * LMAX)) * DT0
             ARG1 = R * R0 / (2. * ALPHA * T0)
             ARG3 = -(R - R0) ** 2 / (4. * ALPHA * T0)
790          SUM = SUM + T0 ** -1.5 * BESI0E(ARG1) * EXP(ARG3)
             W(J, I, K) = CONST1 * R0 * DR0 * SUM * DT0 / (1. * LMAX)
            END IF
770       CONTINUE
760     CONTINUE
750   CONTINUE
      RETURN
      END


      REAL FUNCTION WGHT(J, I, K, DT, RAD0, W)
C.....FUNCTION TO GIVE THE PROPER WEIGHT FUNCTION VALUE
C.....WARNING: GOOD ONLY FOR DTSHORT = 0.1 AND DTLONG = 1.0 !!!!!!!
      PARAMETER (PI = 3.14159265358979)
      INTEGER J, I, K
      REAL DT, RAD0, W(78, 78, 10)
      REAL ALPHA, DT0, T0, CONST1, DR0, R0, R, ARG1, ARG2, ARG3
      INTEGER NUMWTI
C.....ALPHA: THERMAL DIFFUSIVITY OF SOLID; DT0: RECOLLECTION TIME INTERVAL;
C.....T0: RECOLLECTION TIME; CONST1: ALGEBRAIC EXPRESSION FROM GREEN'S FUNCTION;
C.....DR0: DUMMY RADIAL POSITION INTERVAL; R0: DUMMY RADIAL POSITION;
C.....R: RADIAL POSITION; ARG1, ARG2, ARG3: GREEN'S FUNCTION ARGUMENTS;
C.....NUMWTI: NUMBER OF TIME STEPS TO BE PRECALCULATED
      REAL ERF, BESI0E
C.....ERF: ERROR FUNCTION; BESI0E: EXPONENTIAL BESSEL FUNCTION I0(ARG) X EXP(-ARG)
C.....(FROM LINKED LIBRARY)
      IF (DT.EQ.0.1) NUMWTI = 10
      IF (DT.EQ.1.0) NUMWTI = 2
      IF (K.LE.NUMWTI) THEN
        WGHT = W(J, I, K)
        RETURN
      END IF
      ALPHA = 5.79E-07
      DT0 = DT
      CONST1 = (4 * PI * ALPHA) ** -.5
      T0 = (K - .5) * DT0
      IF (I .LE. 48) THEN
        DR0 = RAD0 / 12.
        R0 = RAD0 * (I - .5) / 12.
       ELSEIF (I .LE. 66) THEN
        DR0 = RAD0 / 6.
        R0 = RAD0 * (4. + ((I - 48) - .5) / 6.)
       ELSEIF (I .LE. 75) THEN
        DR0 = RAD0 / 3.
        R0 = RAD0 * (7. + ((I - 66) - .5) / 3.)
       ELSE
        DR0 = RAD0
```

```
        R0 = RAD0 * .10. + ((I - 75) - .5))
      END IF
      IF (J .LE. 48) THEN
        R = RAD0 * (J - .5) / 12.
       ELSEIF (J .LE. 66) THEN
        R = RAD0 * (4. + ((J - 48) - .5) / 6.)
       ELSEIF (J .LE. 75) THEN
        R = RAD0 * (7. + ((J - 66) - .5) / 3.)
       ELSE
        R = RAD0 * (10. + ((J - 75) - .5))
      END IF
      ARG1 = R * R0 / (2. * ALPHA * T0)
      IF (I .EQ. J) THEN
        ARG2 = (DR0 / 2.) / (4. * ALPHA * T0) ** .5
        WGHT = R * T0 ** -1 * BESI0E(ARG1) * ERF(ARG2) * DT0
       ELSE
        ARG3 = -(R - R0) ** 2 / (4. * ALPHA * T0)
        WGHT = CONST1 * R0 * T0 ** -1.5 * BESI0E(ARG1) * EXP(ARG3) * DR0 * DT0
      END IF
      RETURN
      END


      REAL FUNCTION XI (TI, PSAT)
C.....FUNCTION TO GIVE XI GIVEN TEMPERATURE
C.....XI: MOLAR FRACTION OF WATER VAPOR
      REAL TI, PSAT(0:100)
      INTEGER TSAT
      REAL TVAL, XA, XB
      TSAT = -1
      TVAL = TI
      IF (TVAL .GT. 100.) TVAL = 100.
  900 CONTINUE
        TSAT = TSAT + 1
      IF (TSAT .LT. TVAL - 1.) GOTO 900
      XA = PSAT(TSAT) / 1.01325
      XB = PSAT(TSAT + 1) / 1.01325
C.....LINEARLY INTERPOLATE XI
      XI = XA + (XB - XA) * (TI - TSAT)
      RETURN
      END
```

Appendix C Material Properties as a Function of Temperature

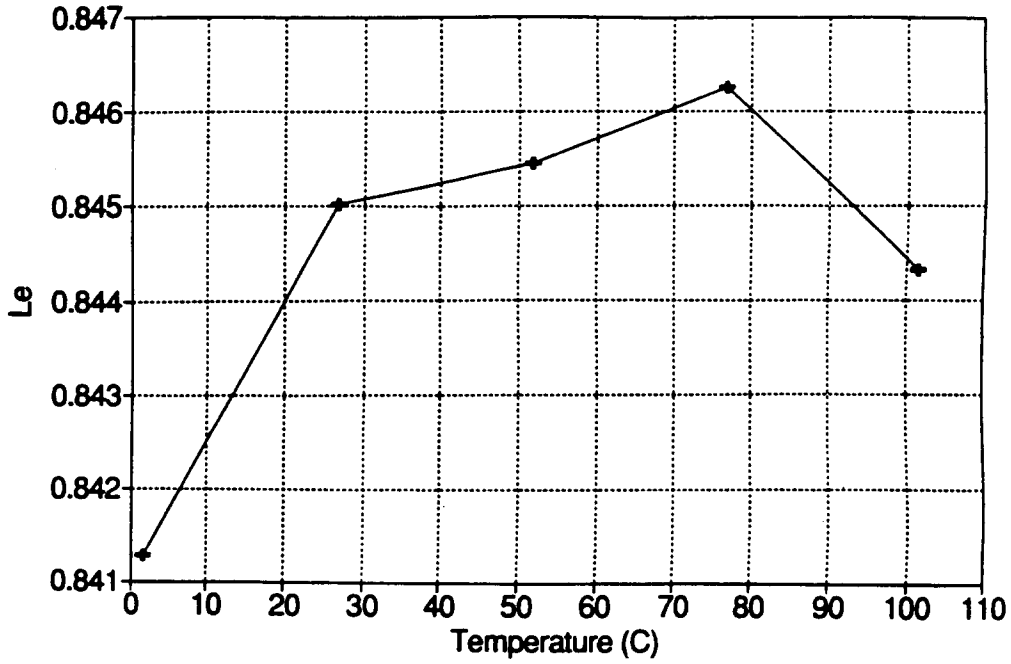

FIGURE 48
Lewis Number of Steam in Air vs. Temperature

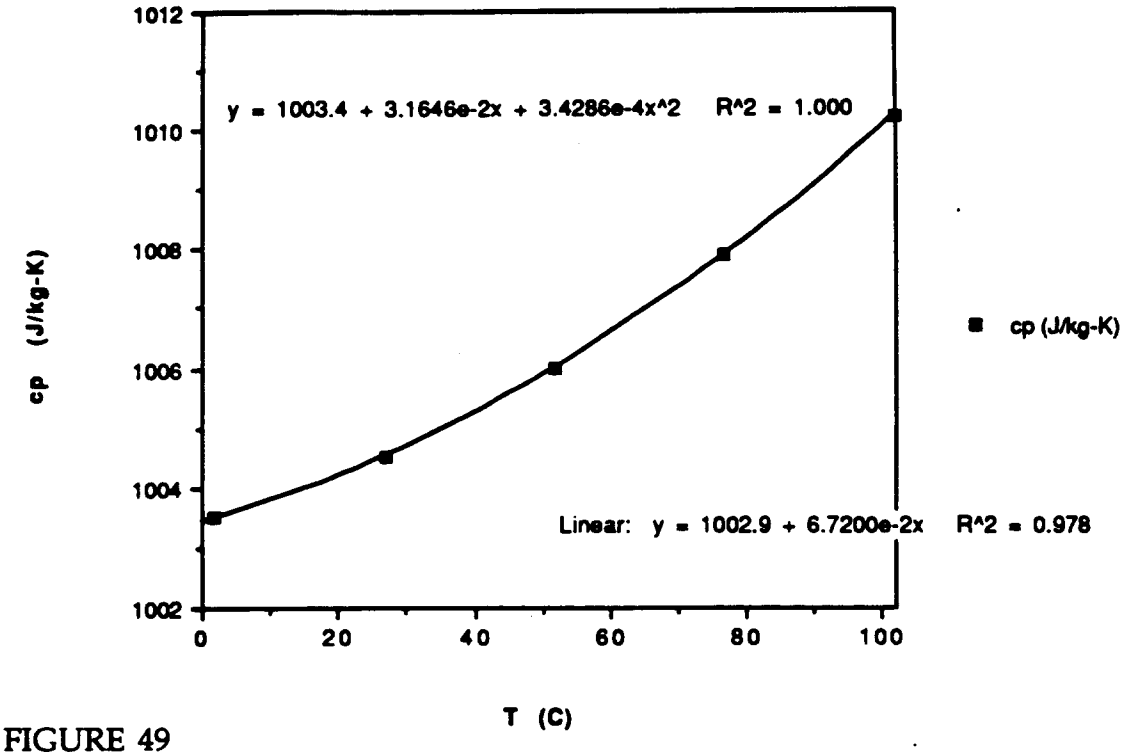

FIGURE 49
Constant Pressure Specific Heat for Air $c_{p,air}$ vs. Temperature

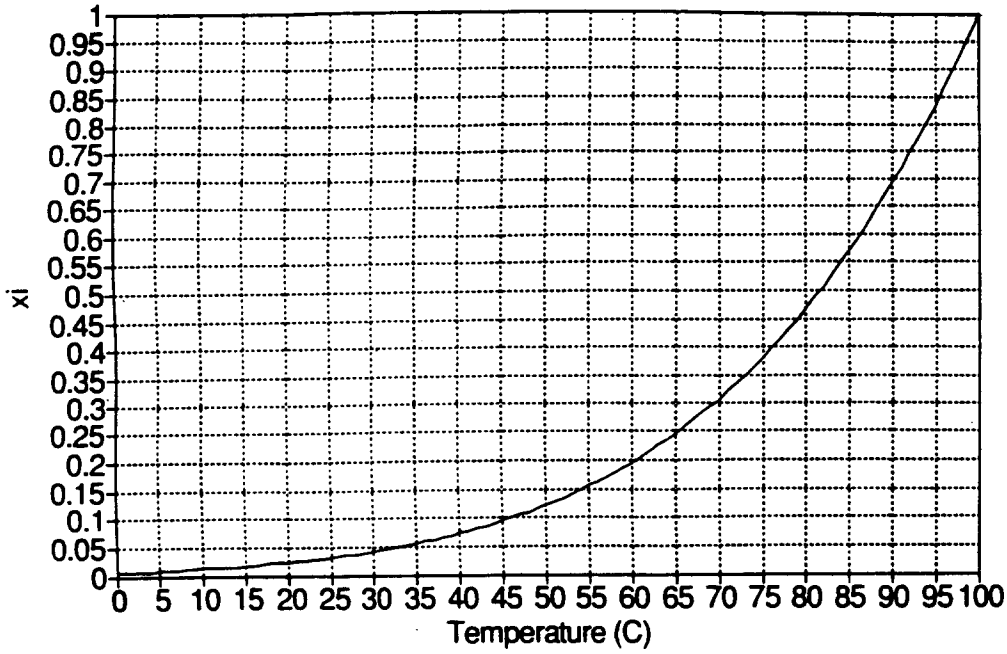

FIGURE 50
Molar Fraction of Water Vapor at Liquid-Vapor Interface x_i vs. Temperature

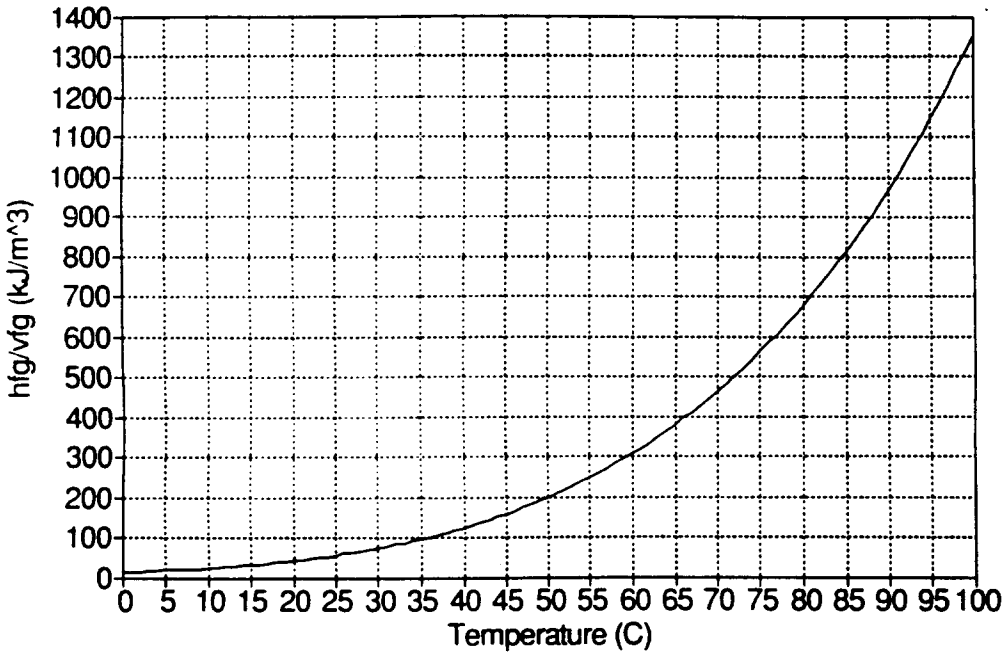

FIGURE 51
Ratio of Λ/v_{fg} for Water vs. Temperature

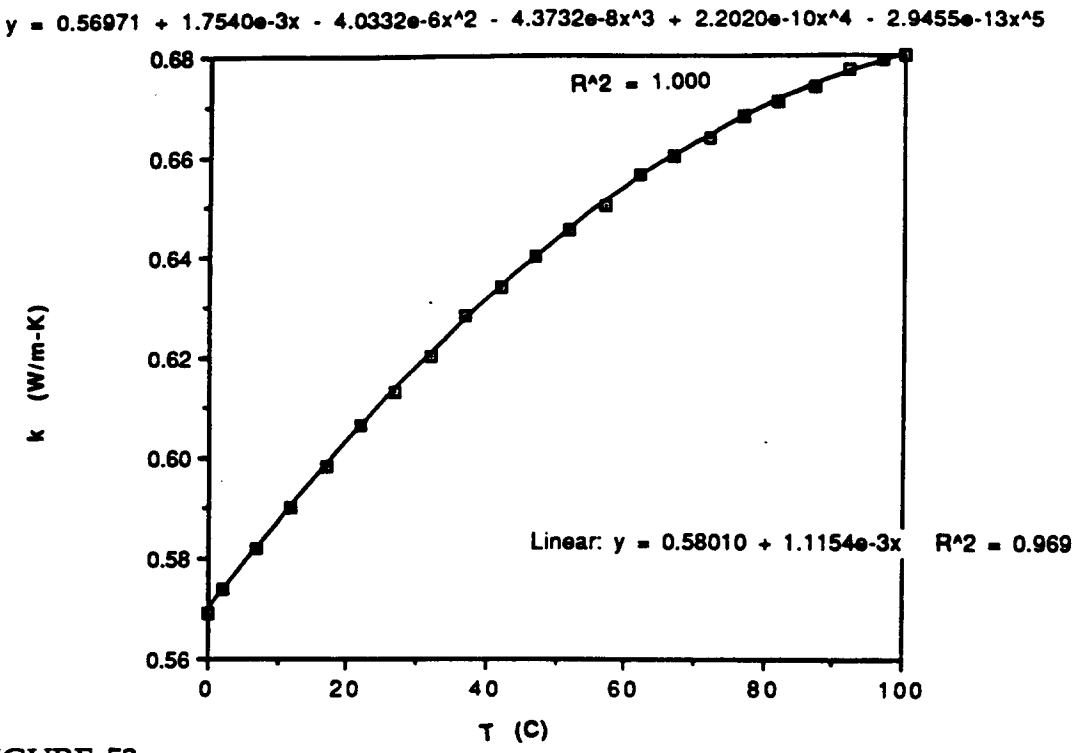

FIGURE 52
Thermal Conductivity of Water k_l vs. Temperature

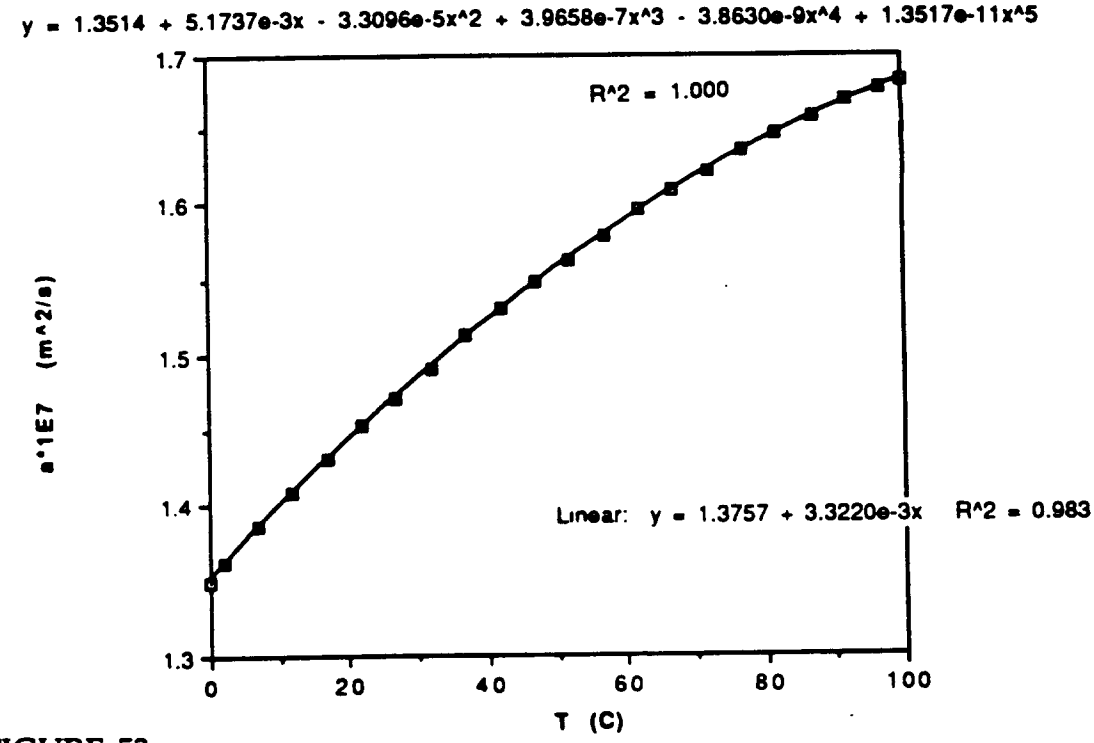

FIGURE 53
Thermal Diffusivity of Water α_l vs. Temperature

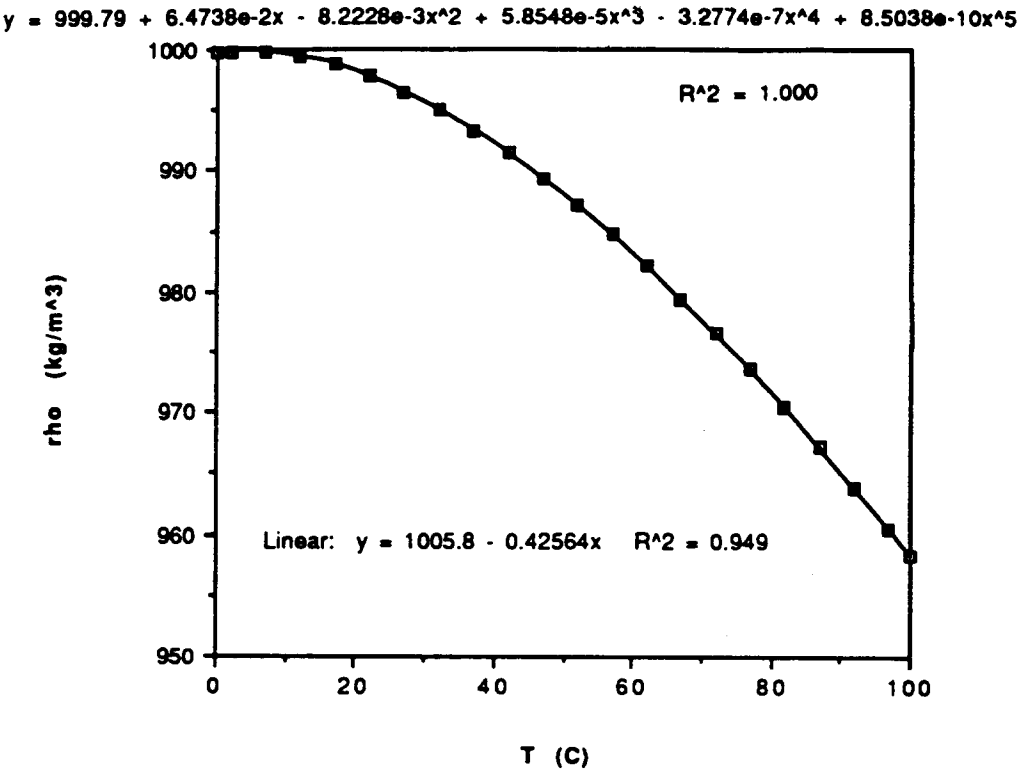

FIGURE 54
Density of Water ρ_l vs. Temperature

TABLE 2
Thermodynamic Properties of Saturated Water [23]

t (C)	P (MPa)	xi	10^3 vf (m^3/kg)	rho (kg/m^3)	10^3 vg (m^3/kg)	vfg (m^3/kg)	hfg (kJ/kg)	hfg/vfg (kJ/m^3)
0	0.0006109	0.0060291	1.0002	999.800	206278	206.2770	2501.4	12.1264
1	0.0006567	0.0064811	1.0002	999.800	192577	192.5760	2499.0	12.9767
2	0.0007056	0.0069637	1.0001	999.900	179889	179.8880	2496.7	13.8792
3	0.0007577	0.0074779	1.0001	999.900	168132	168.1310	2494.3	14.8355
4	0.0008131	0.0080247	1.0001	999.900	157232	157.2310	2491.9	15.8487
5	0.0008721	0.0086070	1.0001	999.900	147120	147.1190	2489.6	16.9224
6	0.0009349	0.0092267	1.0001	999.900	137734	137.7330	2487.2	18.0581
7	0.0010016	0.0098850	1.0002	999.800	129017	129.0160	2484.8	19.2596
8	0.0010724	0.0105838	1.0002	999.800	120917	120.9160	2482.5	20.5308
9	0.0011477	0.0113269	1.0003	999.700	113386	113.3850	2480.1	21.8733
10	0.0012276	0.0121155	1.0004	999.600	106379	106.3780	2477.7	23.2915
11	0.0013123	0.0129514	1.0004	999.600	99857	99.8560	2475.4	24.7897
12	0.0014022	0.0138386	1.0005	999.500	93784	93.7830	2473.0	26.3694
13	0.0014974	0.0147782	1.0007	999.300	88124	88.1230	2470.7	28.0369
14	0.0015983	0.0157740	1.0008	999.201	82848	82.8470	2468.3	29.7935
15	0.0017051	0.0168280	1.0009	999.101	77926	77.9250	2465.9	31.6445
16	0.0018181	0.0179433	1.0011	998.901	73333	73.3320	2463.6	33.5952
17	0.0019376	0.0191226	1.0012	998.801	69044	69.0430	2461.2	35.6474
18	0.0020640	0.0203701	1.0014	998.602	65038	65.0370	2458.8	37.8062
19	0.0021975	0.0216876	1.0016	998.403	61293	61.2920	2456.5	40.0786
20	0.002339	0.023084	1.0018	998.203	57791	57.7900	2454.1	42.4658
21	0.002487	0.024545	1.0020	998.004	54514	54.5130	2451.8	44.9764
22	0.002645	0.026104	1.0022	997.805	51447	51.4460	2449.4	47.6111
23	0.002810	0.027733	1.0024	997.606	48574	48.5730	2447.0	50.3778
24	0.002985	0.029460	1.0027	997.307	45883	45.8820	2444.7	53.2823
25	0.003169	0.031276	1.0029	997.108	43360	43.3590	2442.3	56.3274
26	0.003363	0.033190	1.0032	996.810	40994	40.9930	2439.9	59.5199
27	0.003567	0.035204	1.0035	996.512	38774	38.7730	2437.6	62.8685
28	0.003782	0.037325	1.0037	996.314	36690	36.6890	2435.2	66.3741
29	0.004008	0.039556	1.0040	996.016	34733	34.7320	2432.8	70.0449
30	0.004246	0.041905	1.0043	995.718	32894	32.8930	2430.5	73.8911
31	0.004496	0.044372	1.0046	995.421	31165	31.1640	2428.1	77.9136
32	0.004759	0.046968	1.0050	995.025	29540	29.5390	2425.7	82.1186
33	0.005034	0.049682	1.0053	994.728	28011	28.0100	2423.4	86.5191
34	0.005324	0.052544	1.0056	994.431	26571	26.5700	2421.0	91.1178
35	0.005628	0.055544	1.0060	994.036	25216	25.2150	2418.6	95.9191
36	0.005947	0.058692	1.0063	993.739	23940	23.9390	2416.2	100.932
37	0.006281	0.061989	1.0067	993.345	22737	22.7360	2413.9	106.171
38	0.006632	0.065453	1.0071	992.950	21602	21.6010	2411.5	111.638
39	0.006999	0.069075	1.0074	992.654	20533	20.5320	2409.1	117.334
40	0.007384	0.072874	1.0078	992.260	19523	19.5220	2406.7	123.281
41	0.007786	0.076842	1.0082	991.867	18570	18.5690	2404.3	129.479
42	0.008208	0.081007	1.0086	991.473	17671	17.6700	2401.9	135.931

TABLE 2 (Continued)
Thermodynamic Properties of Saturated Water [23]

t (C)	P (MPa)	xi	10^3 vf (m^3/kg)	rho (kg/m^3)	10^3 vg (m^3/kg)	vfg (m^3/kg)	hfg (kJ/kg)	hfg/vfg (kJ/m^3)
43	0.008649	0.085359	1.0090	991.080	16821	16.8200	2399.5	142.658
44	0.009111	0.089919	1.0095	990.589	16018	16.0170	2397.2	149.666
45	0.009593	0.094676	1.0099	990.197	15258	15.2570	2394.8	156.964
46	0.010098	0.099660	1.0103	989.805	14540	14.5390	2392.4	164.551
47	0.010624	0.104851	1.0108	989.315	13861	13.8600	2390.0	172.439
48	0.011175	0.110289	1.0112	988.924	13218	13.2170	2387.6	180.646
49	0.011749	0.115954	1.0117	988.435	12609	12.6080	2385.2	189.182
50	0.012349	0.121875	1.0121	988.045	12032	12.0310	2382.7	198.047
51	0.012975	0.128053	1.0126	987.557	11485	11.4840	2380.3	207.271
52	0.013628	0.134498	1.0131	987.069	10968	10.9670	2377.9	216.823
53	0.014309	0.141219	1.0136	986.582	10476	10.4750	2375.5	226.778
54	0.015019	0.148226	1.0141	986.096	10011	10.0100	2373.1	237.073
55	0.015758	0.155519	1.0146	985.610	9568	9.5670	2370.7	247.800
56	0.016529	0.163129	1.0151	985.125	9149	9.1480	2368.2	258.877
57	0.017331	0.171044	1.0156	984.640	8751	8.7500	2365.8	270.378
58	0.018166	0.179284	1.0161	984.155	8372	8.3710	2363.4	282.332
59	0.019036	0.187871	1.0166	983.671	8013	8.0120	2360.9	294.671
60	0.019940	0.196792	1.0172	983.091	7671	7.6700	2358.5	307.497
61	0.020881	0.206079	1.0177	982.608	7346	7.3450	2356.0	320.763
62	0.021860	0.215741	1.0182	982.125	7037	7.0360	2353.6	334.509
63	0.022877	0.225778	1.0188	981.547	6743	6.7420	2351.1	348.725
64	0.023934	0.236210	1.0194	980.969	6463	6.4620	2348.7	363.464
65	0.02503	0.24703	1.0199	980.488	6197	6.1960	2346.2	378.665
66	0.02617	0.25828	1.0205	979.912	5943	5.9420	2343.7	394.431
67	0.02736	0.27002	1.0211	979.336	5701	5.7000	2341.3	410.756
68	0.02859	0.28216	1.0217	978.761	5471	5.4700	2338.8	427.570
69	0.02986	0.29470	1.0222	978.282	5252	5.2510	2336.3	444.927
70	0.03119	0.30782	1.0228	977.708	5042	5.0410	2333.8	462.966
71	0.03256	0.32134	1.0234	977.135	4843	4.8420	2331.4	481.498
72	0.03399	0.33546	1.0240	976.563	4652	4.6510	2328.9	500.734
73	0.03546	0.34996	1.0247	975.895	4470	4.4690	2326.4	520.567
74	0.03699	0.36506	1.0253	975.324	4297	4.2960	2323.9	540.948
75	0.03858	0.38075	1.0259	974.754	4131	4.1300	2321.4	562.086
76	0.04022	0.39694	1.0265	974.184	3973	3.9720	2318.9	583.816
77	0.04192	0.41372	1.0272	973.520	3822	3.8210	2316.3	606.207
78	0.04368	0.43109	1.0278	972.952	3677	3.6760	2313.8	629.439
79	0.04550	0.44905	1.0285	972.290	3539	3.5380	2311.3	653.284
80	0.04739	0.46770	1.0291	971.723	3407	3.4060	2308.8	677.868
81	0.04934	0.48695	1.0298	971.062	3281	3.2800	2306.2	703.116
82	0.05136	0.50688	1.0305	970.403	3160	3.1590	2303.7	729.257
83	0.05345	0.52751	1.0311	969.838	3044	3.0430	2301.1	756.202
84	0.05560	0.54873	1.0318	969.180	2934	2.9330	2298.6	783.711
85	0.05783	0.57074	1.0325	968.523	2828	2.8270	2296.0	812.178

TABLE 2 (Continued)
Thermodynamic Properties of Saturated Water [23]

t (C)	P (MPa)	xi	10^3 vf (m^3/kg)	rho (kg/m^3)	10^3 vg (m^3/kg)	vfg (m^3/kg)	hfg (kJ/kg)	hfg/vfg (kJ/m^3)
86	0.06014	0.59354	1.0332	967.867	2726	2.7250	2293.5	841.662
87	0.06252	0.61702	1.0339	967.212	2629	2.6280	2290.9	871.739
88	0.06498	0.64130	1.0346	966.557	2536	2.5350	2288.3	902.695
89	0.06752	0.66637	1.0353	965.904	2446	2.4450	2285.8	934.901
90	0.07014	0.69223	1.0360	965.251	2361	2.3600	2283.2	967.472
91	0.07284	0.71887	1.0367	964.599	2278	2.2770	2280.6	1001.597
92	0.07564	0.74651	1.0375	963.855	2200	2.1990	2278.0	1035.943
93	0.07852	0.77493	1.0382	963.206	2124	2.1230	2275.4	1071.804
94	0.08149	0.80424	1.0389	962.557	2052	2.0510	2272.8	1108.163
95	0.08455	0.83444	1.0397	961.816	1981.9	1.98086	2270.2	1146.068
96	0.08771	0.86563	1.0404	961.169	1915.0	1.91396	2267.6	1184.769
97	0.09097	0.89780	1.0412	960.430	1850.8	1.84976	2264.9	1224.430
98	0.09433	0.93096	1.0420	959.693	1789.1	1.78806	2262.3	1265.227
99	0.09778	0.96501	1.0427	959.049	1729.9	1.72886	2259.7	1307.048
100	0.10135	1.00025	1.0435	958.313	1672.9	1.67186	2257.0	1349.996

TABLE 3
Thermal Properties of Saturated Water [16]

t (C)	T (K)	10^3 vf (m^3/kg)	rho (kg/m^3)	cp (J/kg-K)	k (W/m-K)	alpha (m^2/s)	alpha (cm^2/s)	alpha*1E (m^2/s)
0	273	1.0002	999.800	4217	0.569	1.350E-07	1.350E-03	1.350
2	275	1.0001	999.900	4211	0.574	1.363E-07	1.363E-03	1.363
7	280	1.0002	999.800	4198	0.582	1.387E-07	1.387E-03	1.387
12	285	1.0005	999.500	4189	0.590	1.409E-07	1.409E-03	1.409
17	290	1.0012	998.801	4184	0.598	1.431E-07	1.431E-03	1.431
22	295	1.0022	997.805	4181	0.606	1.453E-07	1.453E-03	1.453
27	300	1.0035	996.512	4179	0.613	1.472E-07	1.472E-03	1.472
32	305	1.0050	995.025	4178	0.620	1.491E-07	1.491E-03	1.491
37	310	1.0067	993.345	4178	0.628	1.513E-07	1.513E-03	1.513
42	315	1.0086	991.473	4179	0.634	1.530E-07	1.530E-03	1.530
47	320	1.0108	989.315	4180	0.640	1.548E-07	1.548E-03	1.548
52	325	1.0131	987.069	4182	0.645	1.563E-07	1.563E-03	1.563
57	330	1.0156	984.640	4184	0.650	1.578E-07	1.578E-03	1.578
62	335	1.0182	982.125	4186	0.656	1.596E-07	1.596E-03	1.596
67	340	1.0211	979.336	4188	0.660	1.609E-07	1.609E-03	1.609
72	345	1.0240	976.563	4191	0.664	1.622E-07	1.622E-03	1.622
77	350	1.0272	973.520	4195	0.668	1.636E-07	1.636E-03	1.636
82	355	1.0305	970.403	4199	0.671	1.647E-07	1.647E-03	1.647
87	360	1.0339	967.212	4203	0.674	1.658E-07	1.658E-03	1.658
92	365	1.0375	963.855	4209	0.677	1.669E-07	1.669E-03	1.669
97	370	1.0412	960.430	4214	0.679	1.678E-07	1.678E-03	1.678
100	373	1.0435	958.313	4217	0.680	1.683E-07	1.683E-03	1.683

Appendix D.1 Output Data: Effect of Shape Factor β

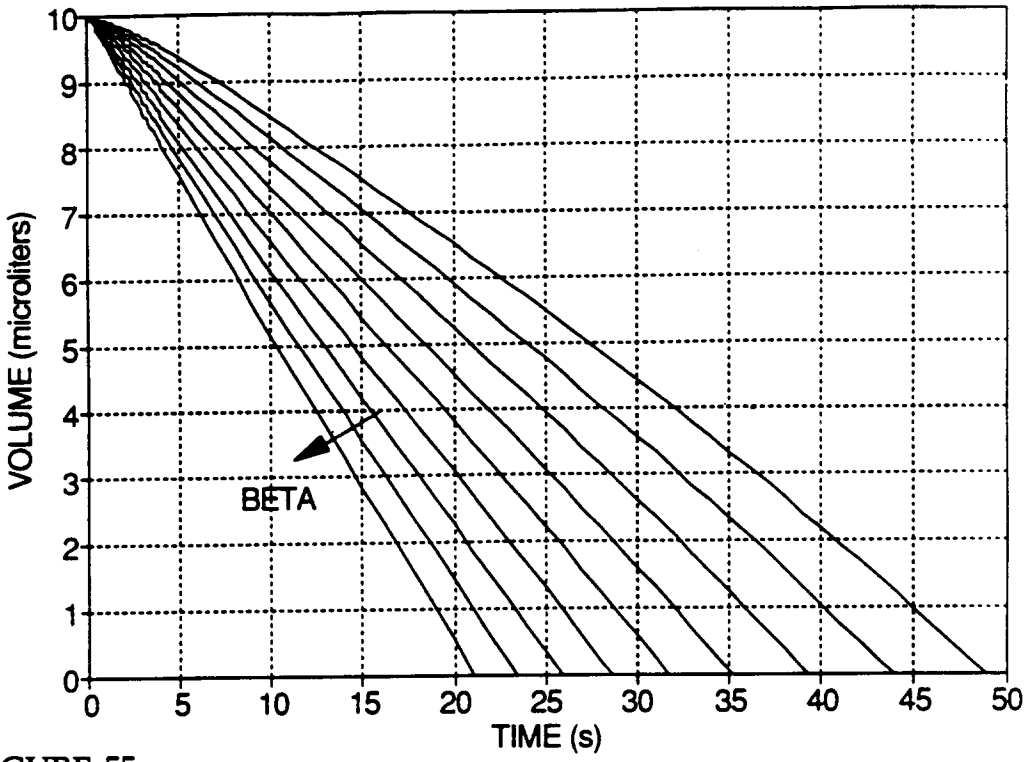

FIGURE 55
Transient Droplet Volume: Runs 1–9 (Effect of β_0)

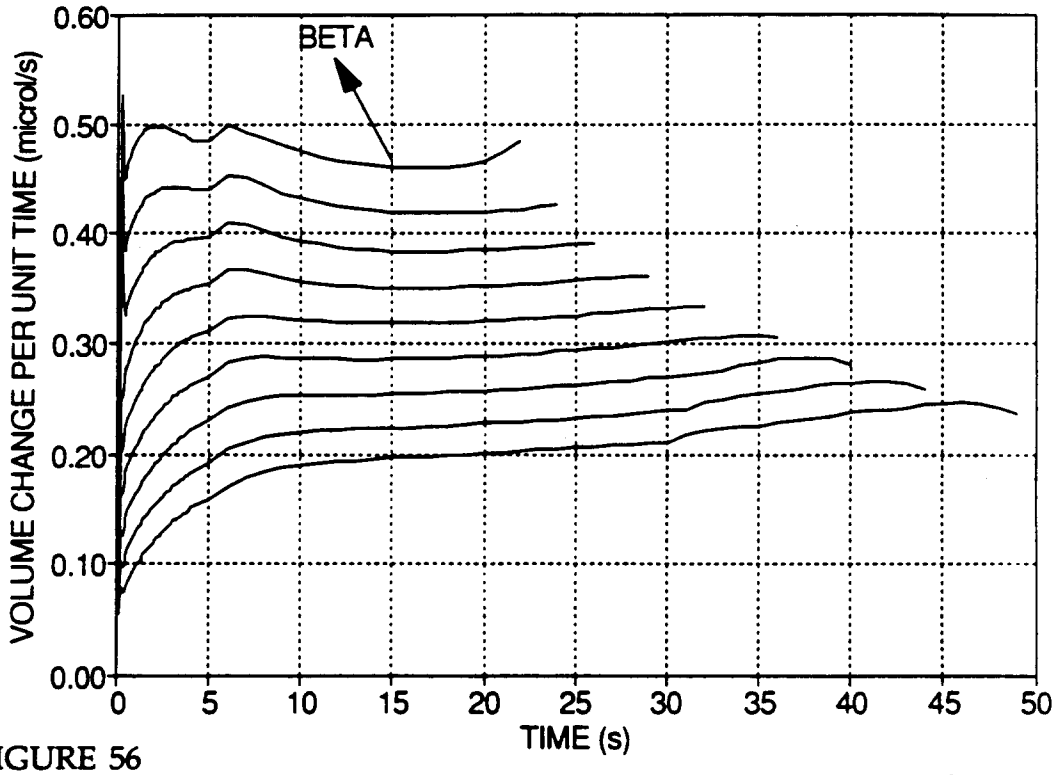

FIGURE 56
Transient Rate of Change of Droplet Volume: Runs 1–9 (Effect of β_0)

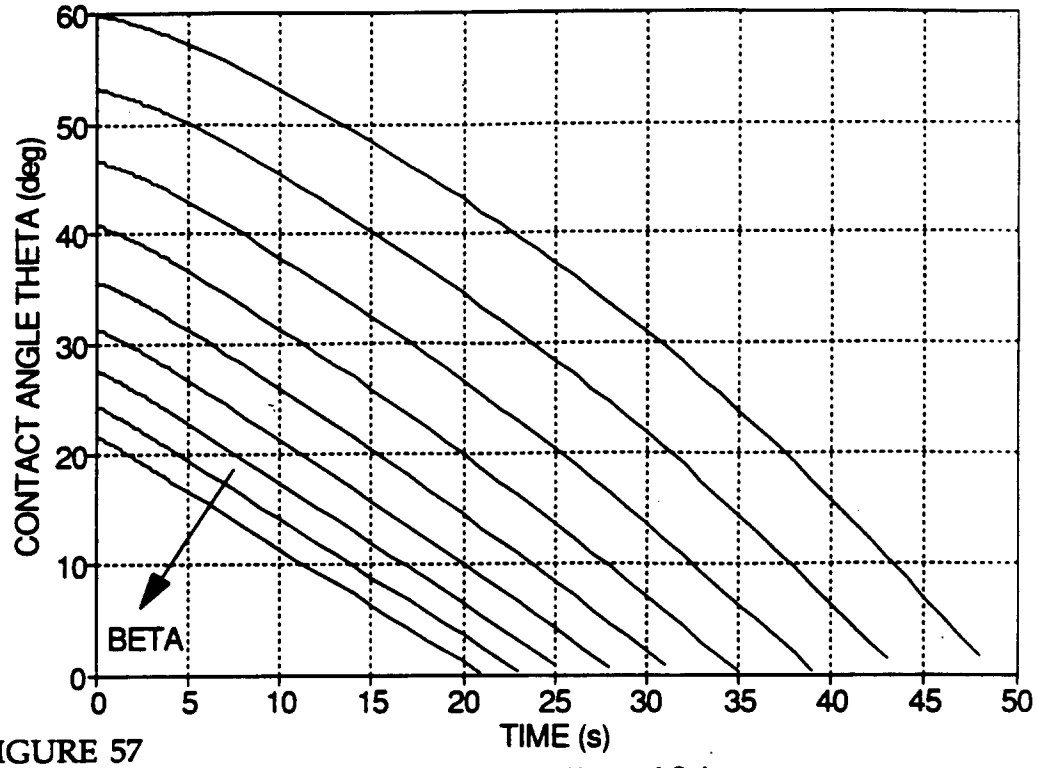

FIGURE 57
Transient Contact Angle θ: Runs 1–9 (Effect of β_0)

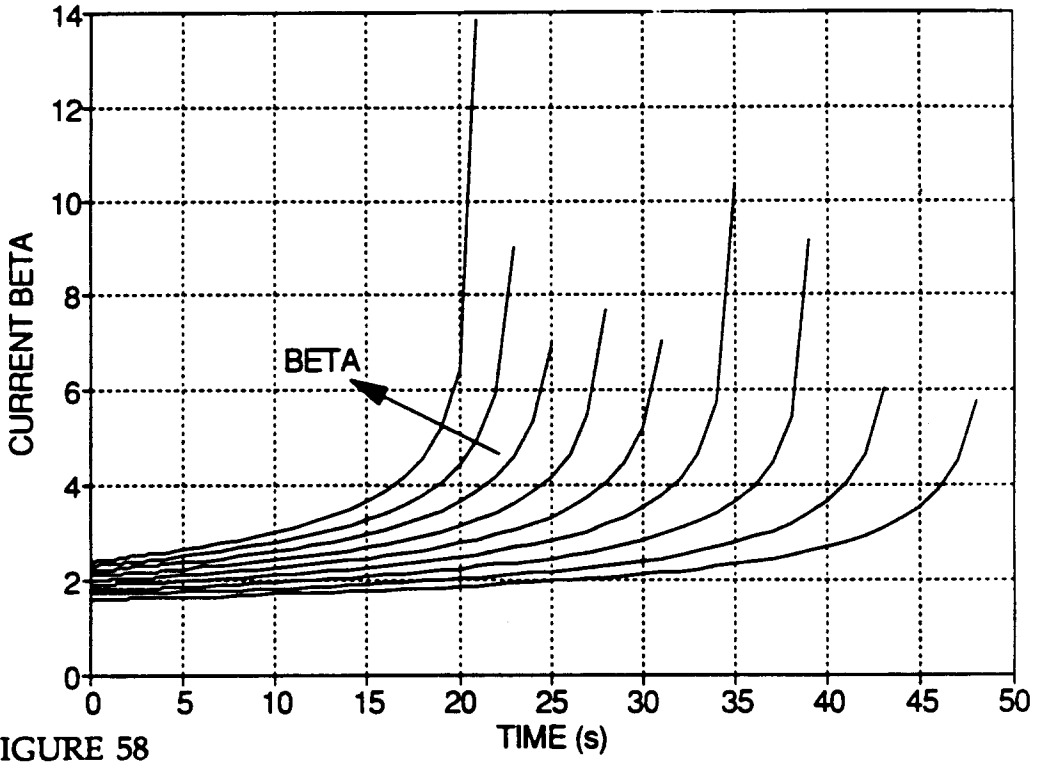

FIGURE 58
Transient Shape Factor β: Runs 1–9 (Effect of β_0)

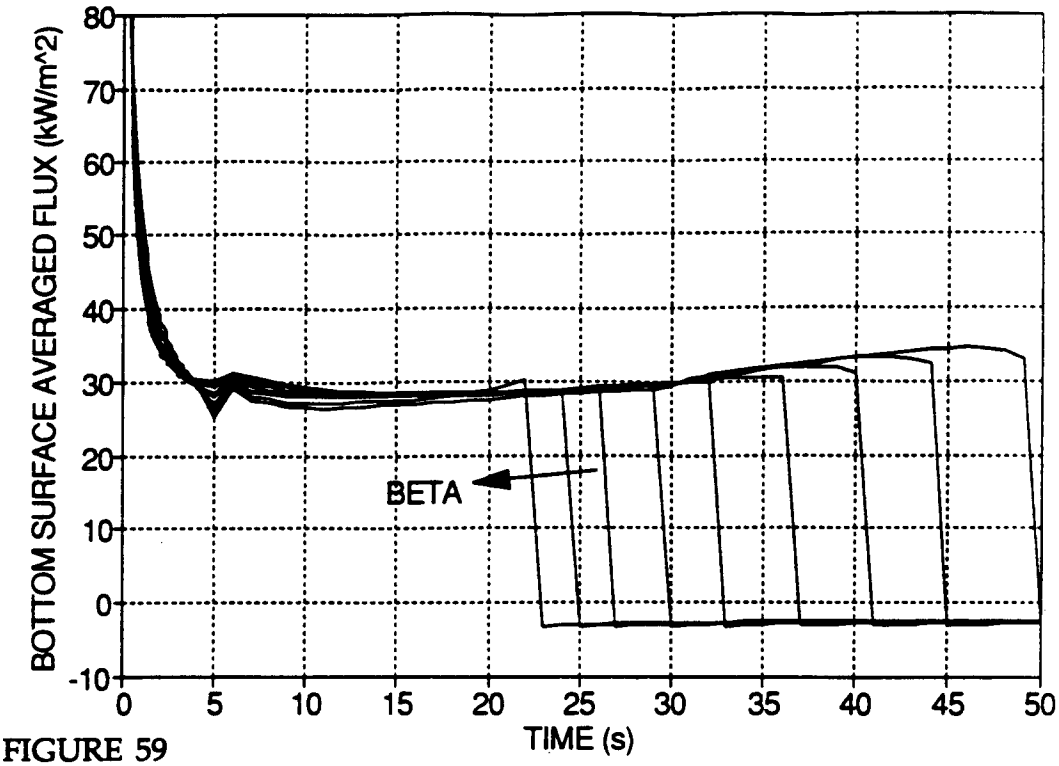

FIGURE 59
Transient Area Averaged Solid-Liquid Flux: Runs 1–9 (Effect of β_0)

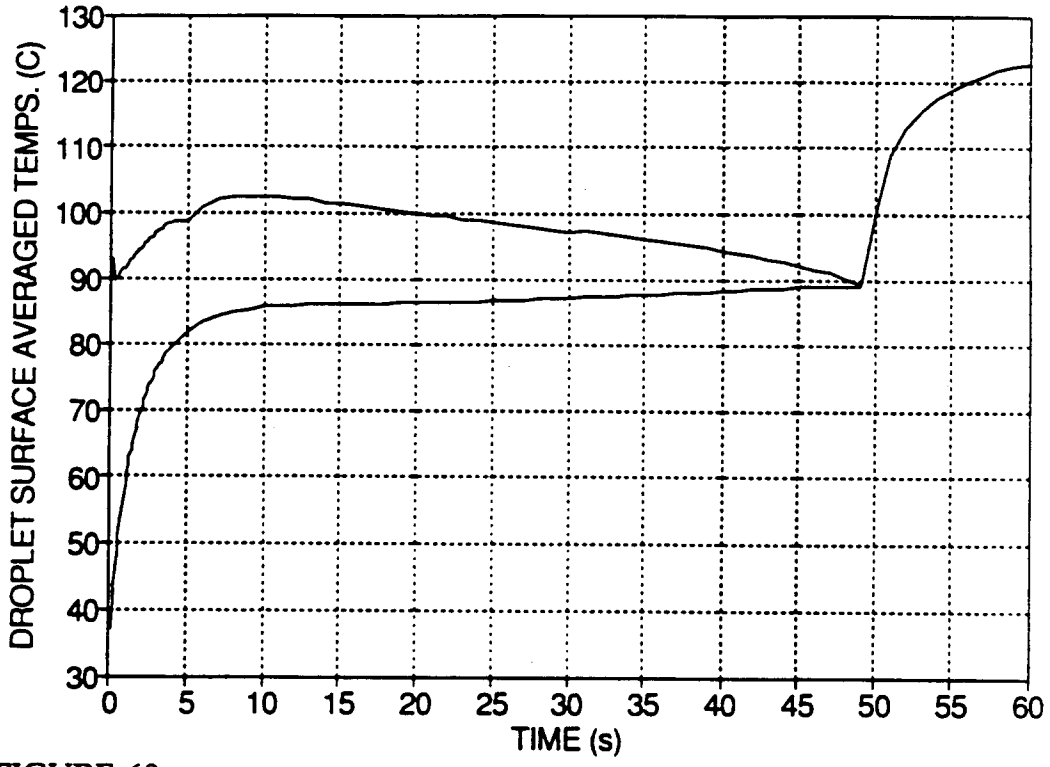

FIGURE 60
Transient Upper and Lower Surface Averaged Droplet Temperatures: Run 1

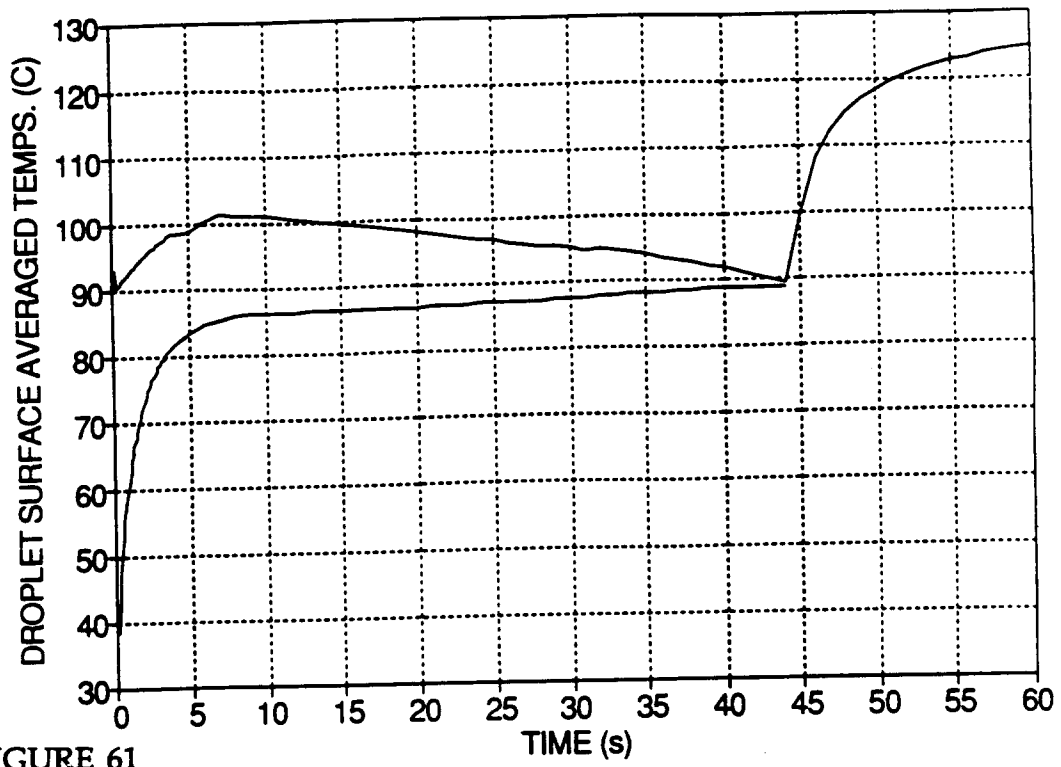

FIGURE 61
Transient Upper and Lower Surface Averaged Droplet Temperatures: Run 2

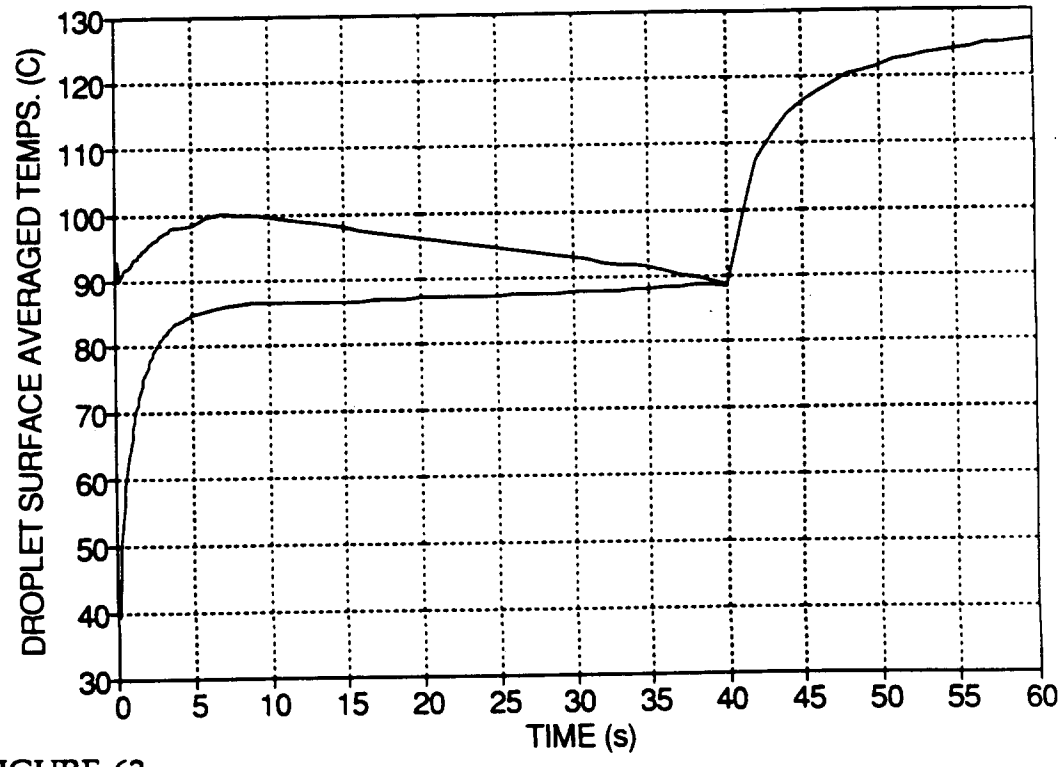

FIGURE 62
Transient Upper and Lower Surface Averaged Droplet Temperatures: Run 3

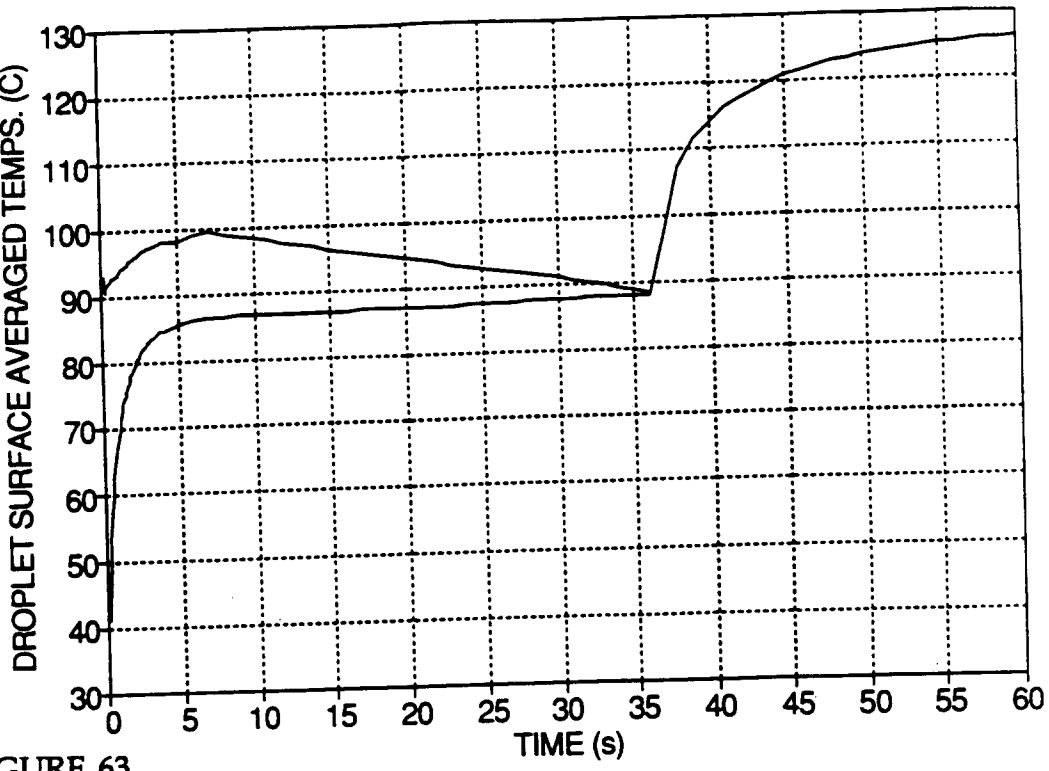

FIGURE 63
Transient Upper and Lower Surface Averaged Droplet Temperatures: Run 4

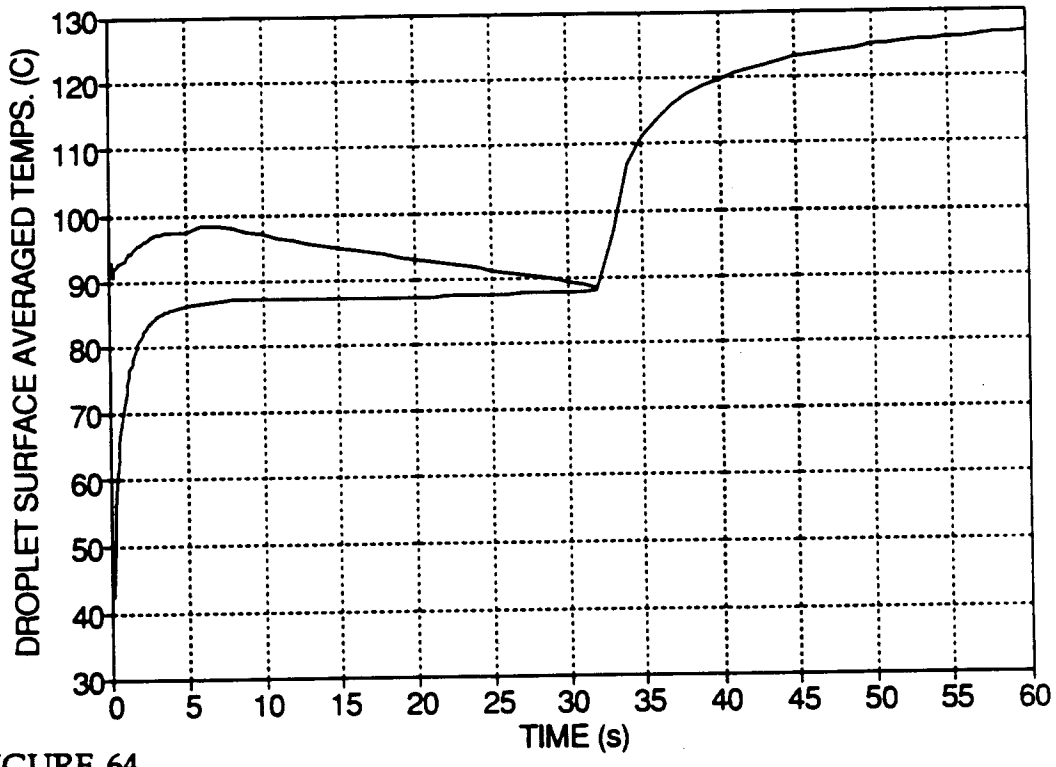

FIGURE 64
Transient Upper and Lower Surface Averaged Droplet Temperatures: Run 5

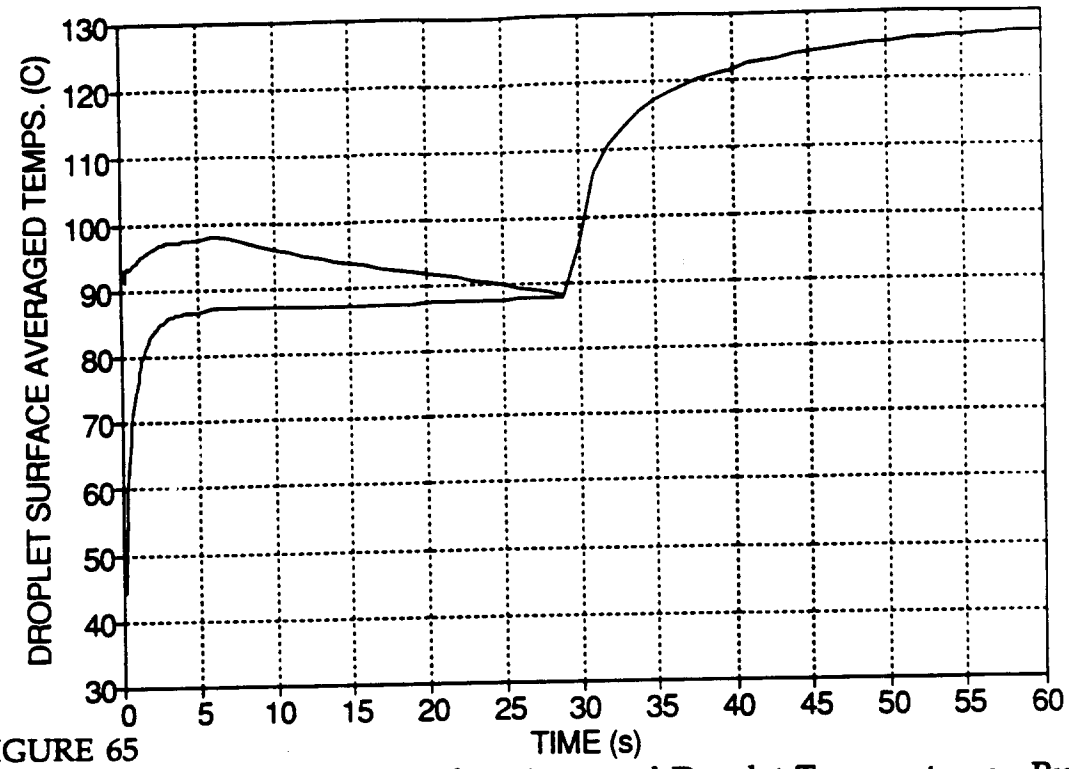

FIGURE 65
Transient Upper and Lower Surface Averaged Droplet Temperatures: Run 6

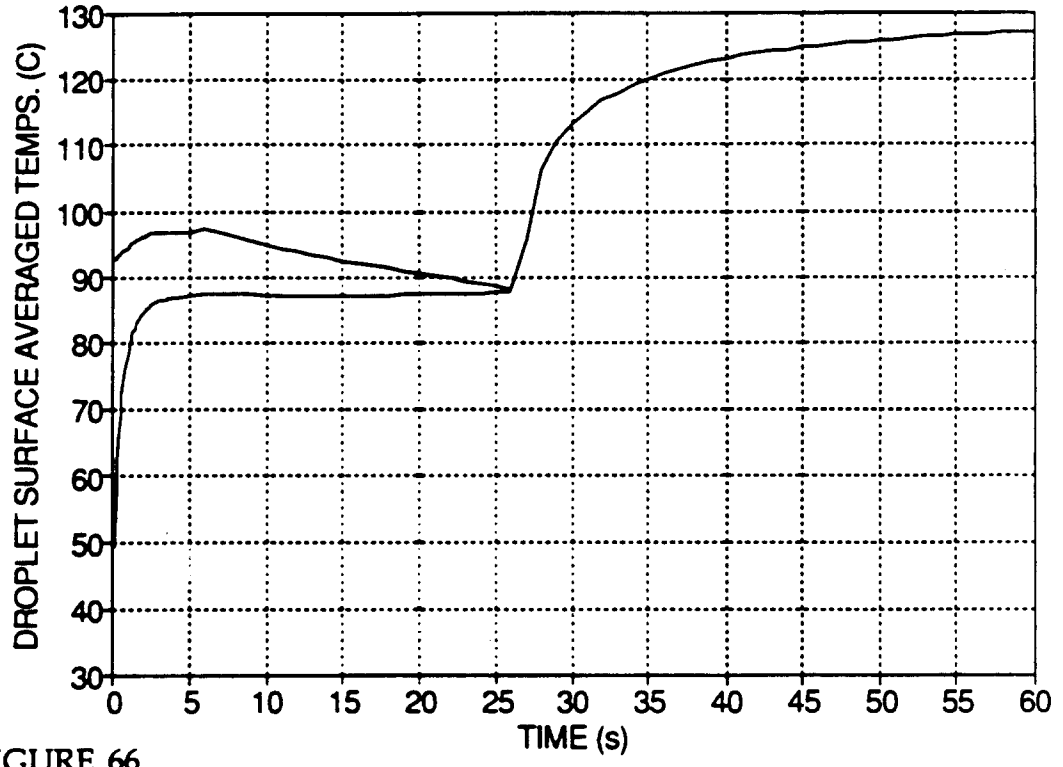

FIGURE 66
Transient Upper and Lower Surface Averaged Droplet Temperatures: Run 7

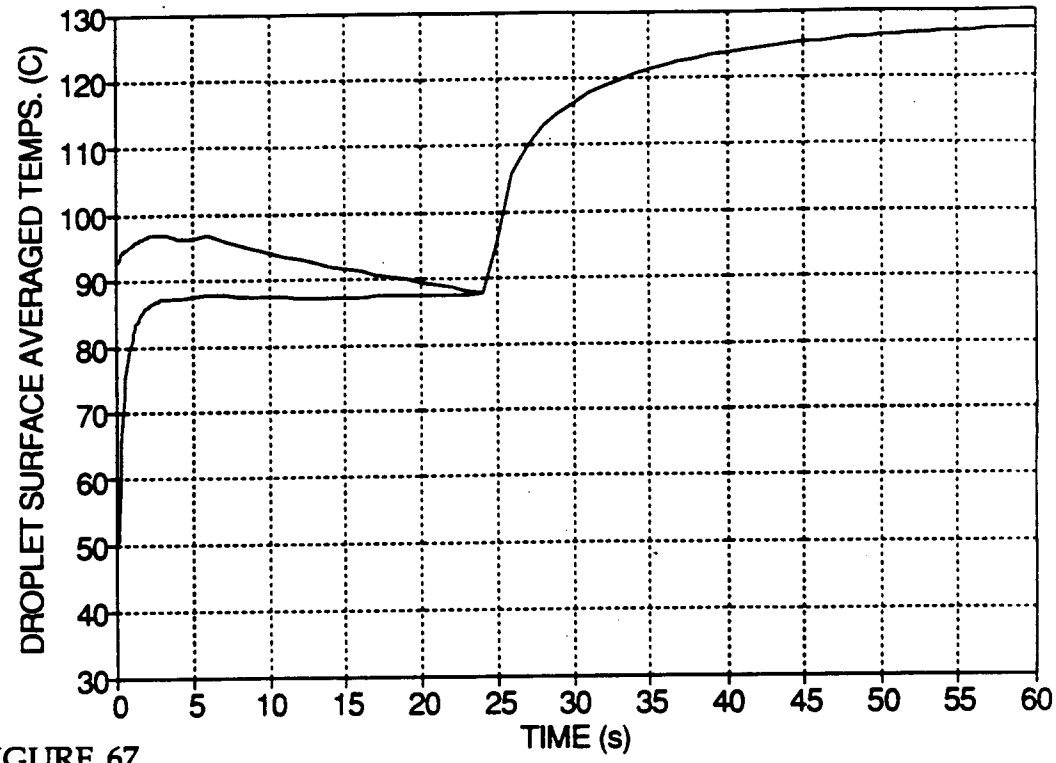

FIGURE 67
Transient Upper and Lower Surface Averaged Droplet Temperatures: Run 8

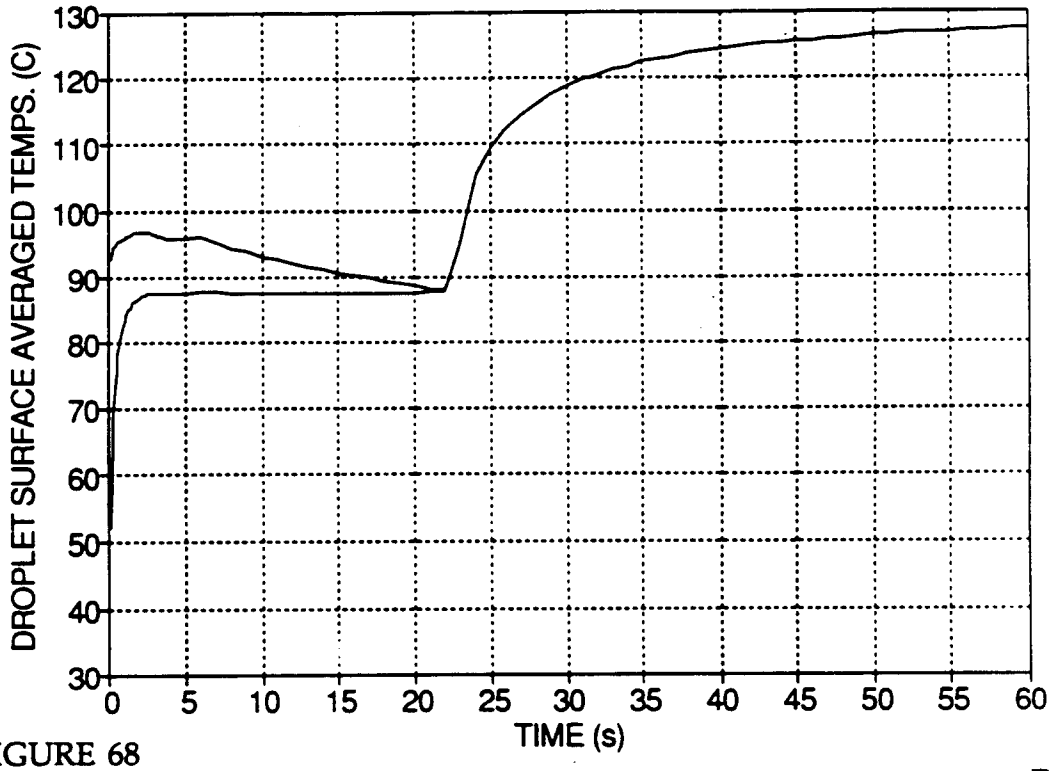

FIGURE 68
Transient Upper and Lower Surface Averaged Droplet Temperatures: Run 9

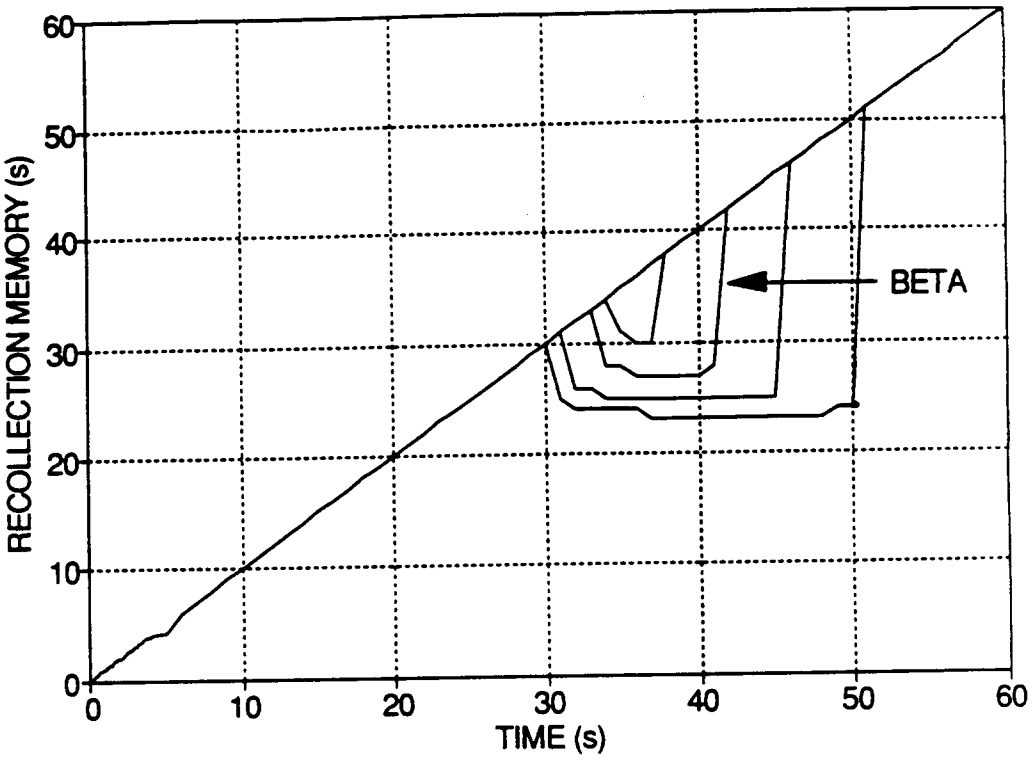

FIGURE 69
Transient Recollection Memory: Runs 1–9 (Effect of β_0)

Appendix D.2 Output Data: Effect of Initial Volume V_0

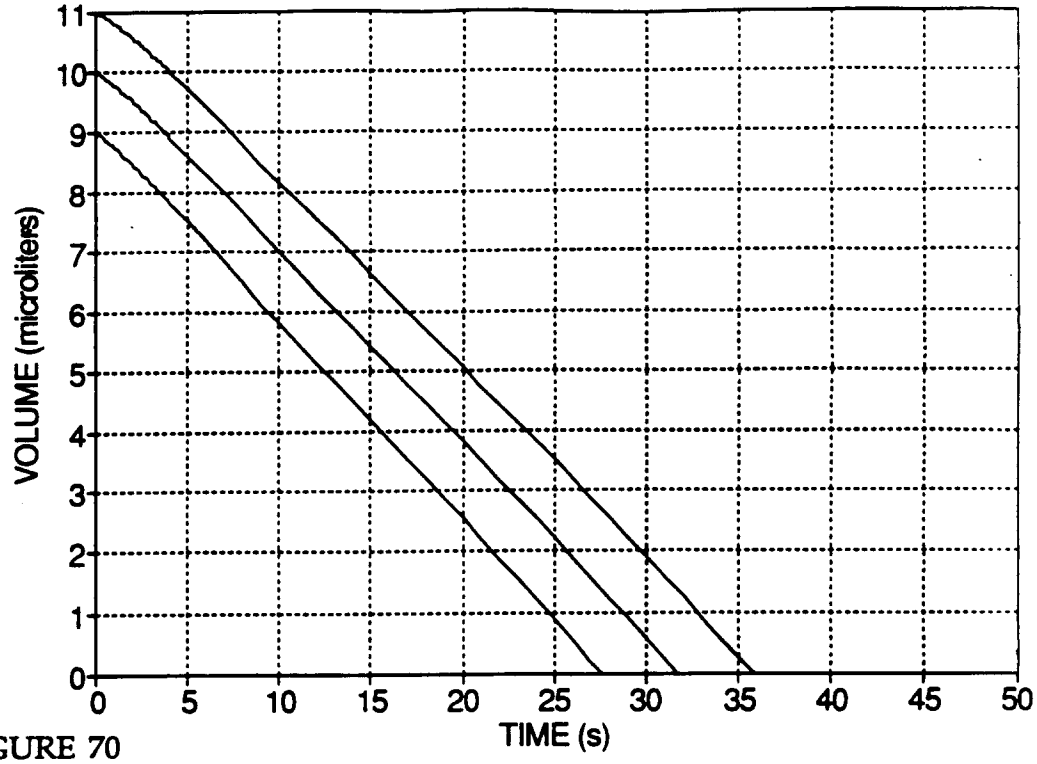

FIGURE 70
Transient Droplet Volume: Runs 5, 10, 11 (Effect of V_0)

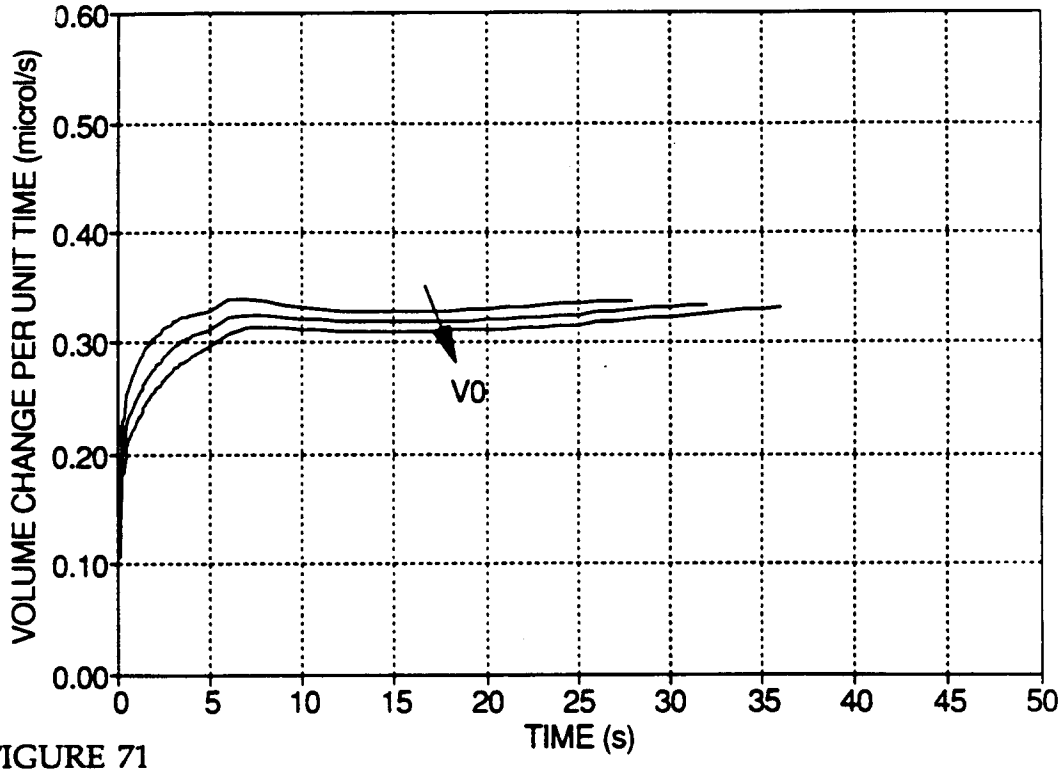

FIGURE 71
Transient Rate of Change of Droplet Volume: Runs 5, 10, 11 (Effect of V_0)

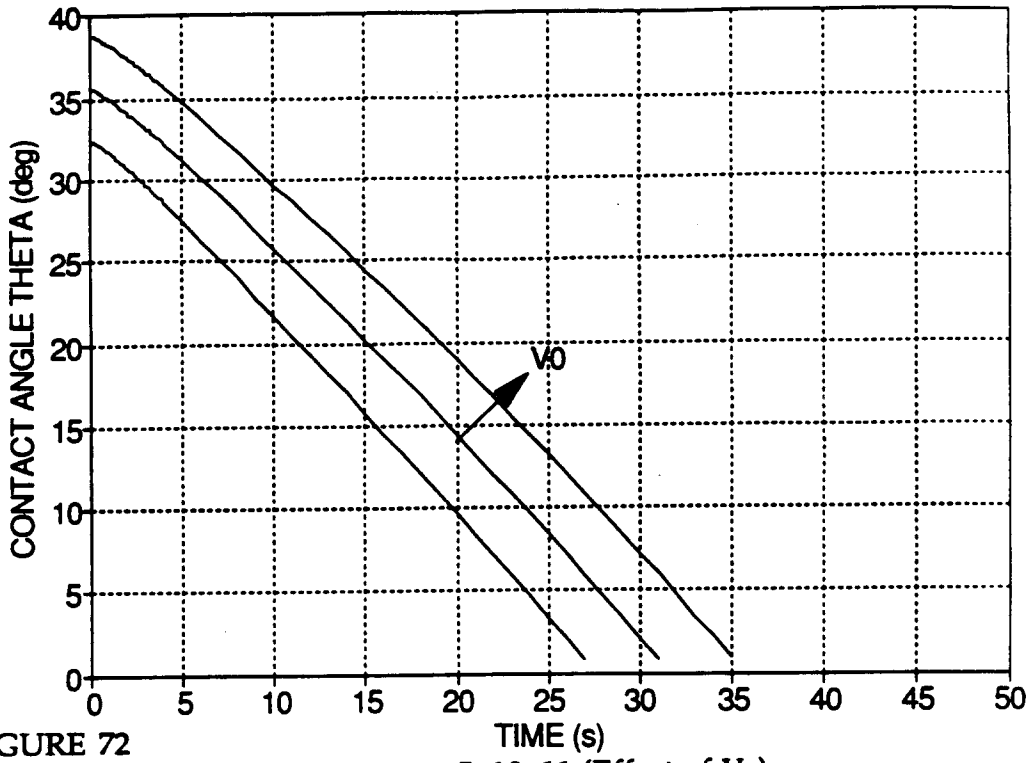

FIGURE 72
Transient Contact Angle θ: Runs 5, 10, 11 (Effect of V_0)

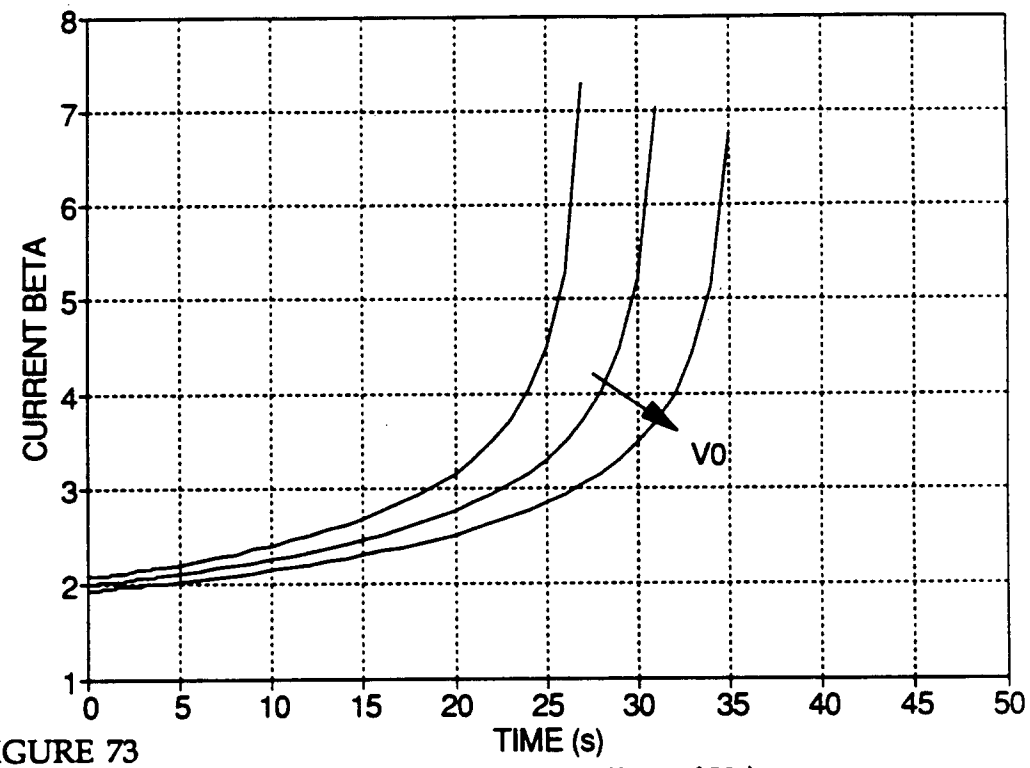

FIGURE 73
Transient Shape Factor β: Runs 5, 10, 11 (Effect of V_0)

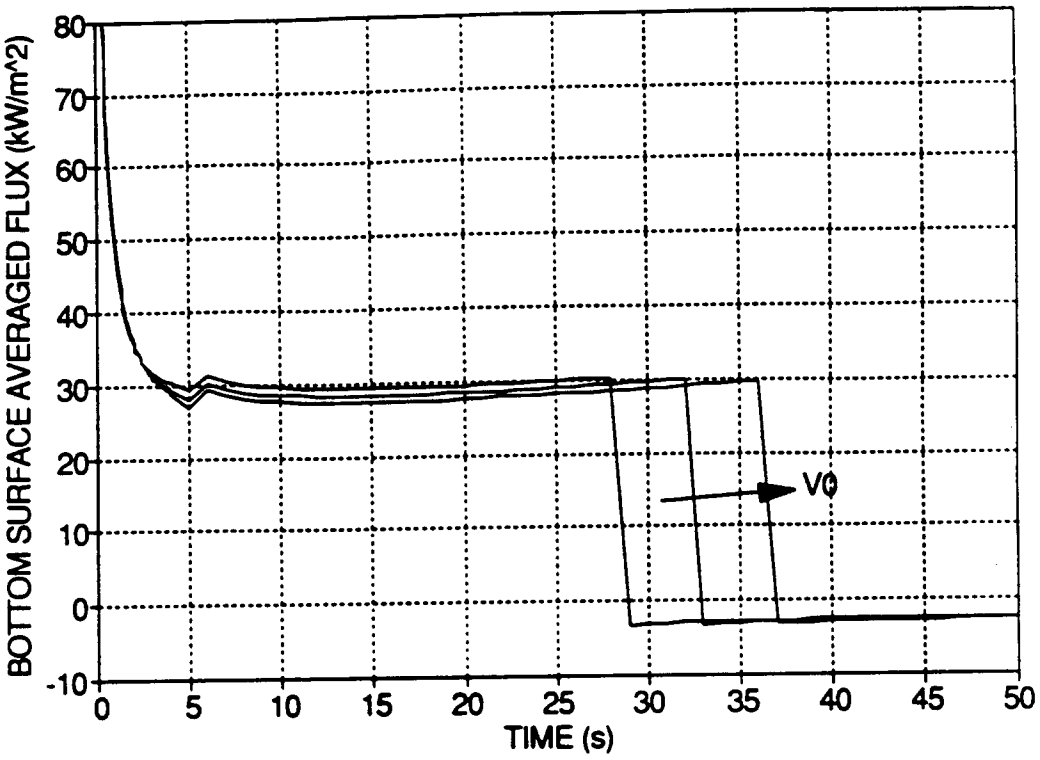

FIGURE 74
Transient Area Averaged Solid-Liquid Flux: Runs 5, 10, 11 (Effect of V_0)

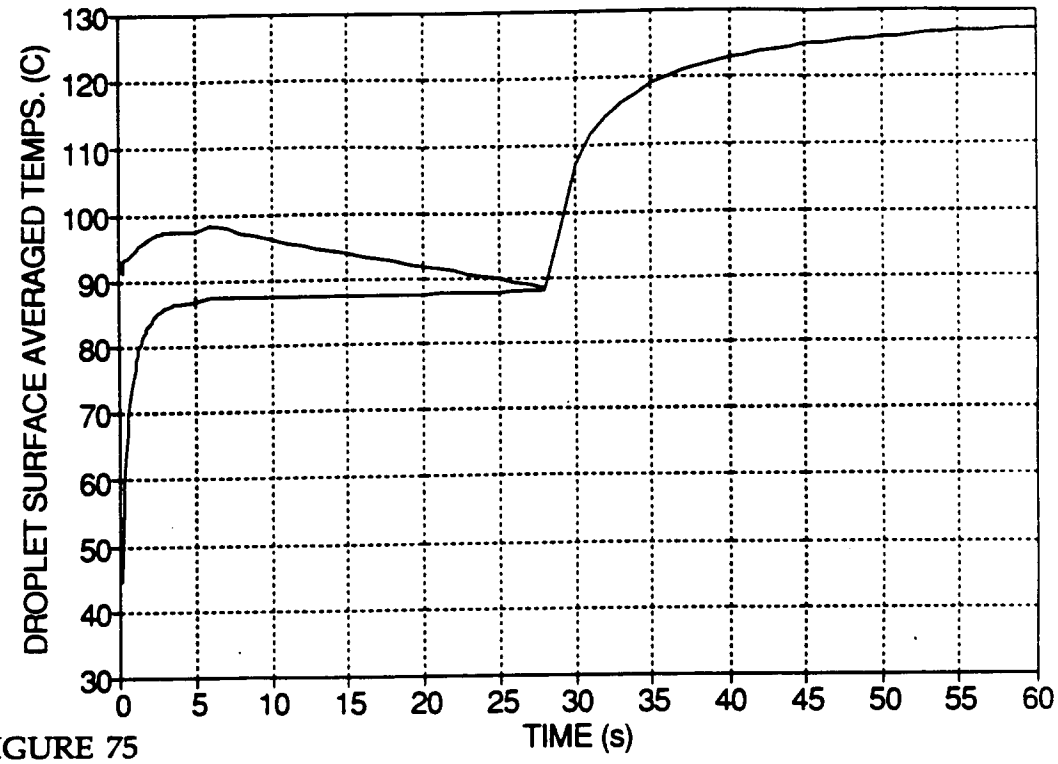

FIGURE 75
Transient Upper and Lower Surface Averaged Droplet Temperatures: Run 10

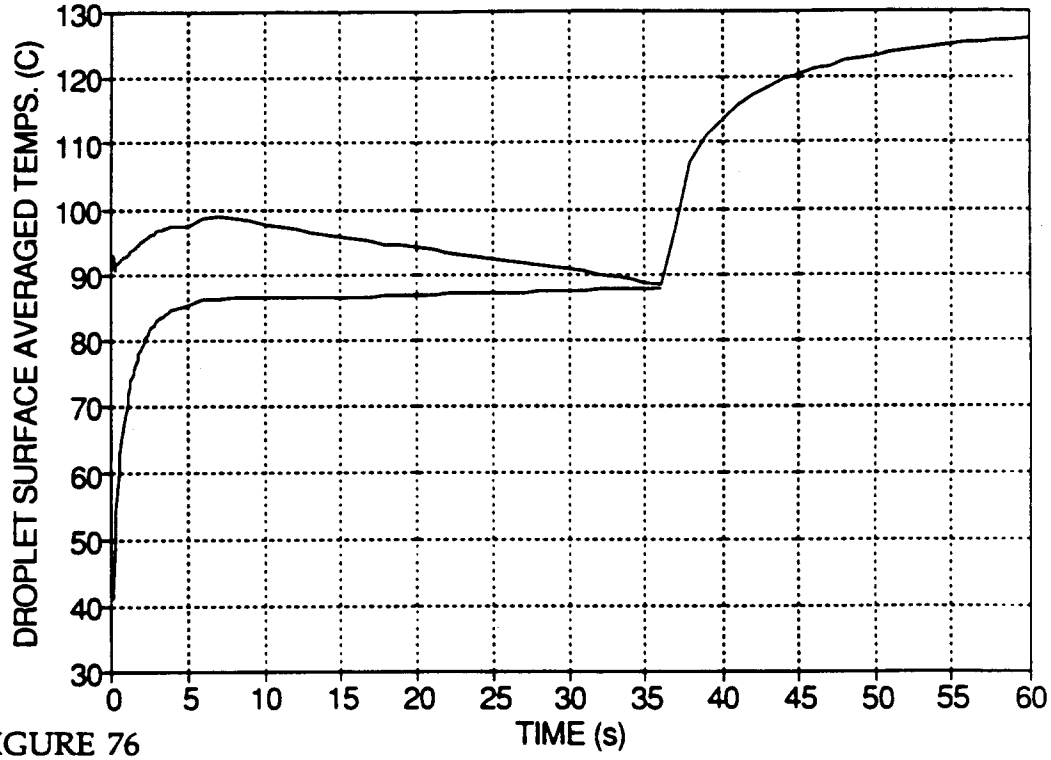

FIGURE 76
Transient Upper and Lower Surface Averaged Droplet Temperatures: Run 11

Appendix D.3 Output Data: Effect of Initial Temperatures

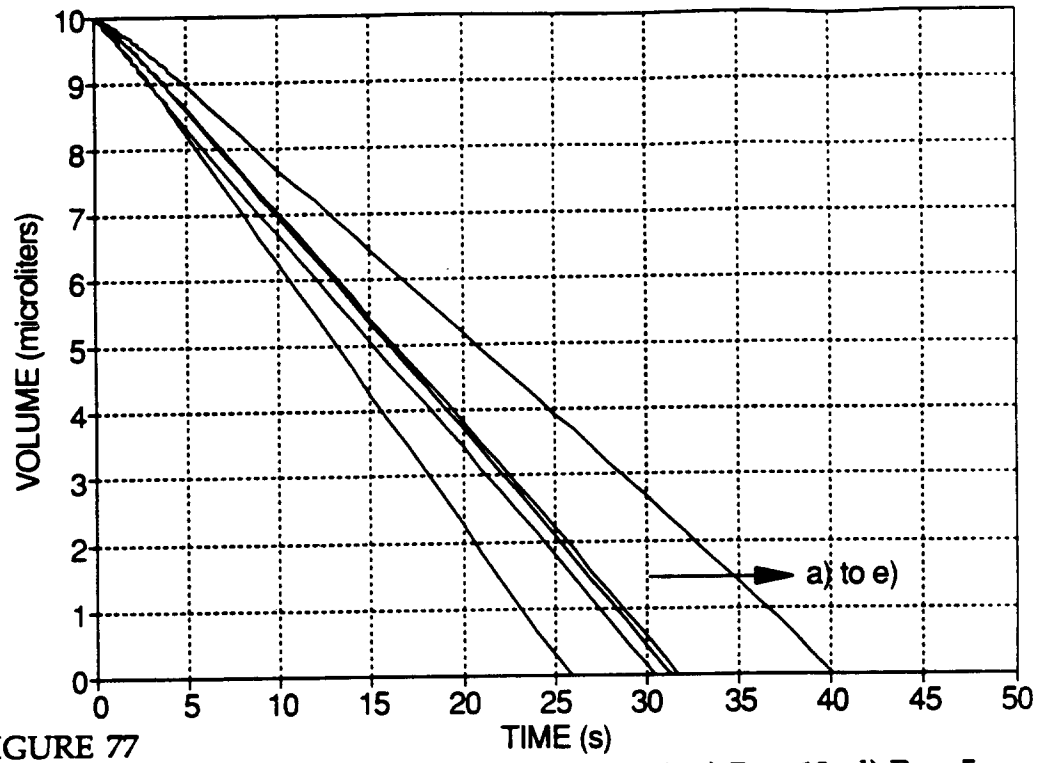

FIGURE 77
Transient Droplet Volume: a) Run 15; b) Run 12; c) Run 13; d) Run 5; e) Run 14 (Effect of Initial Temperatures)

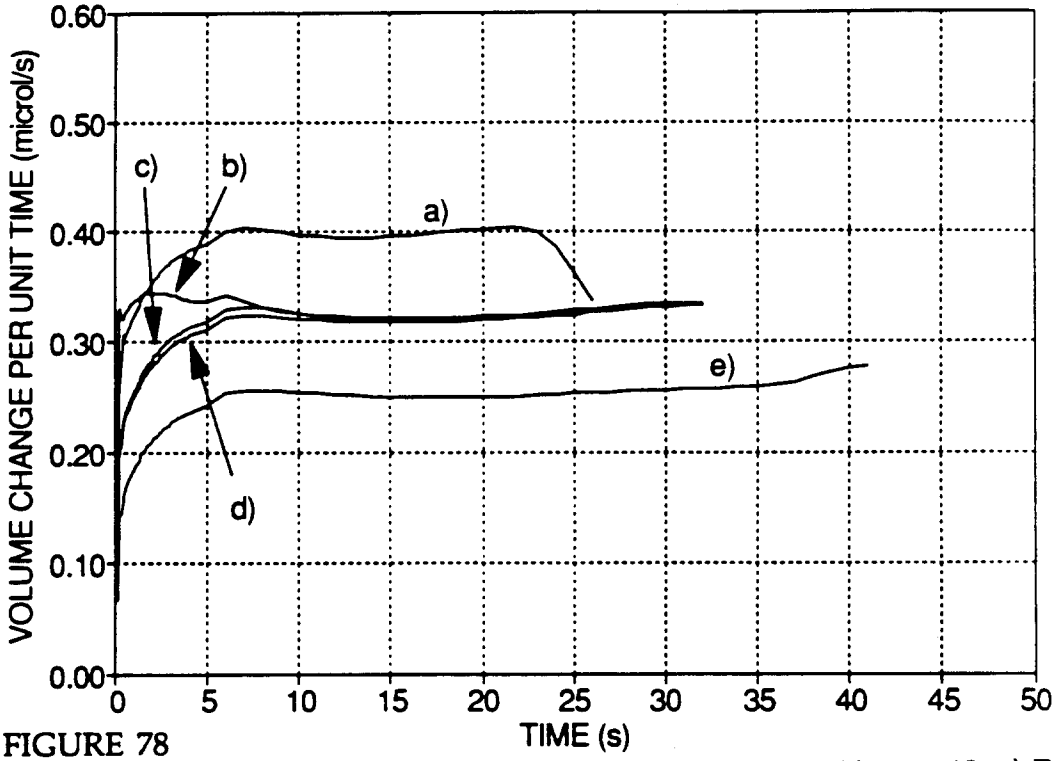

FIGURE 78
Transient Rate of Change of Droplet Volume: a) Run 15; b) Run 12; c) Run 13; d) Run 5; e) Run 14 (Effect of Initial Temperatures)

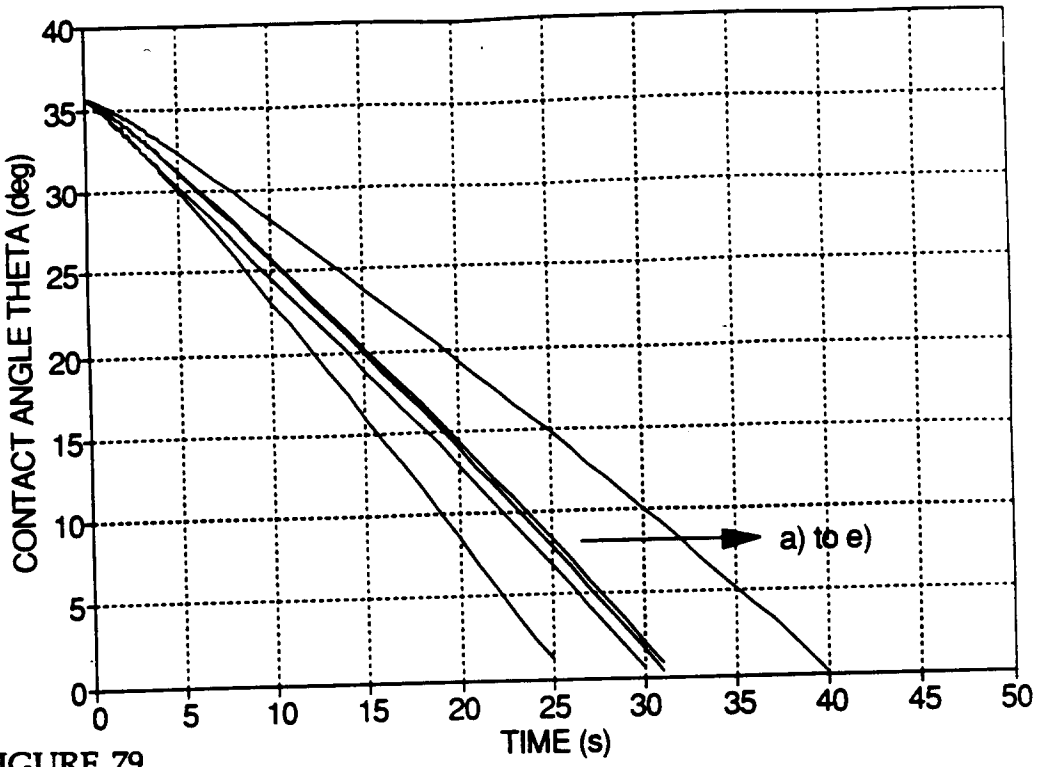

FIGURE 79
Transient Contact Angle θ: a) Run 15; b) Run 12; c) Run 13; d) Run 5; e) Run 14 (Effect of Initial Temperatures)

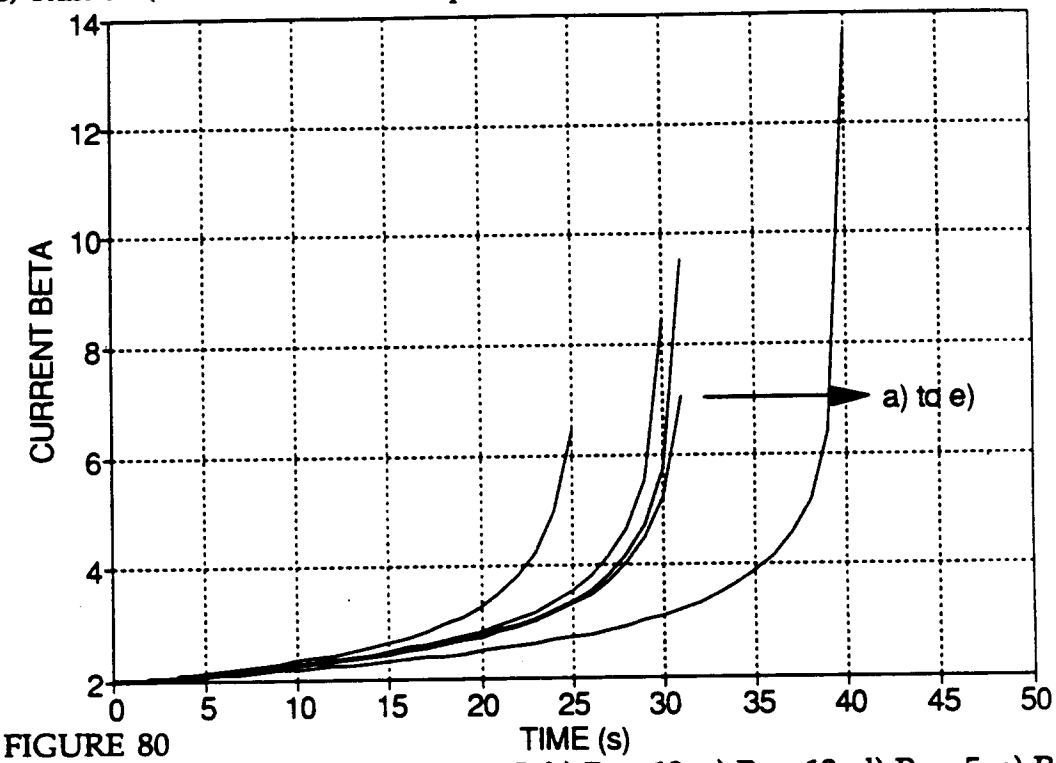

FIGURE 80
Transient Shape Factor β: a) Run 15; b) Run 12; c) Run 13; d) Run 5; e) Run 14 (Effect of Initial Temperatures)

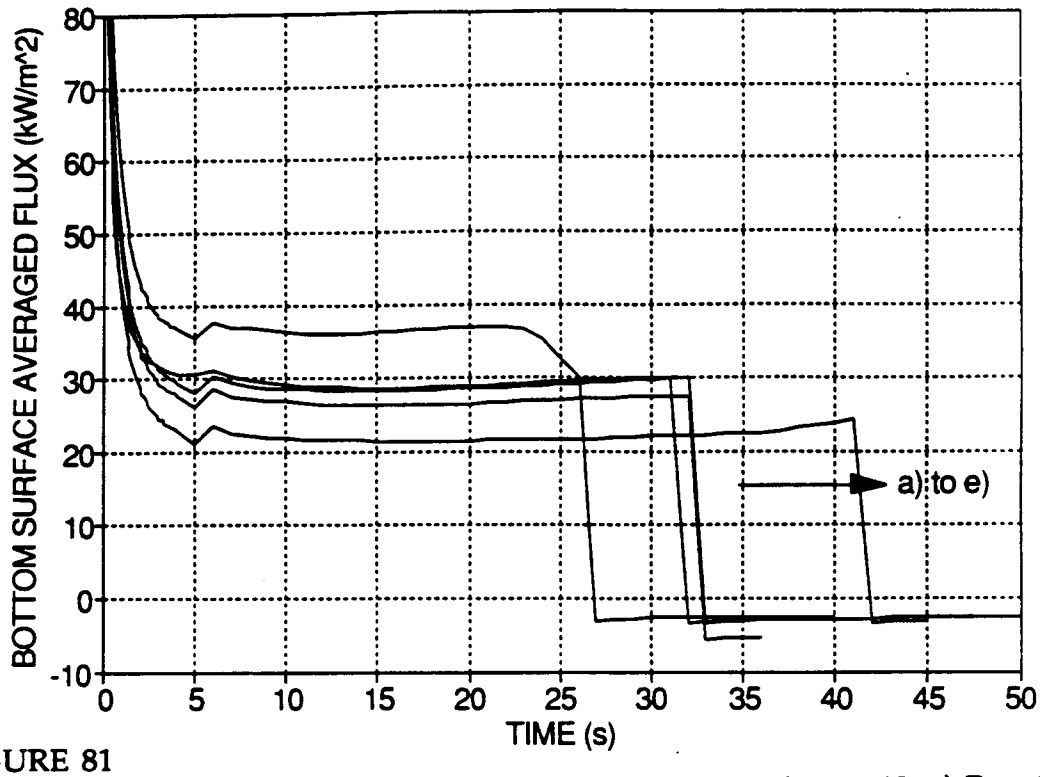

FIGURE 81
Transient Area Averaged Solid-Liquid Flux: a) Run 15; b) Run 12; c) Run 13; d) Run 5; e) Run 14 (Effect of Initial Temperatures)

Appendix D.4 Output Data: Effect of Geometry Model

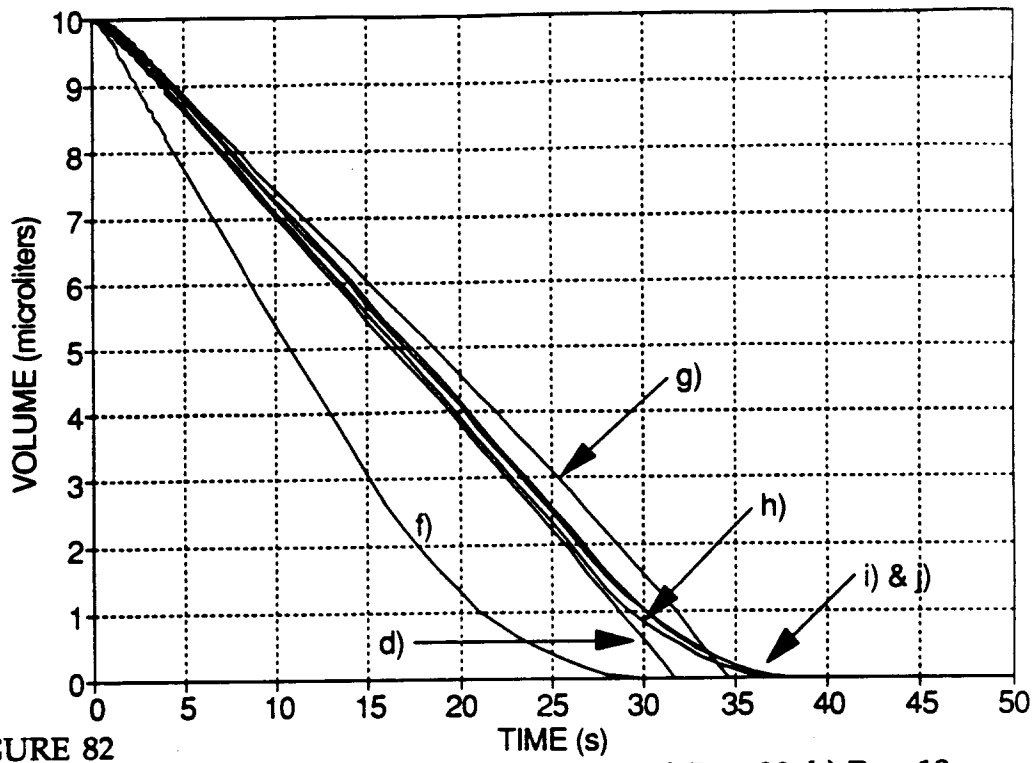

FIGURE 82
Transient Droplet Volume: d) Run 5; f) Run 17; g) Run 20; h) Run 18; i) Run 16; j) Run 19 (Effect of Droplet Geometry)

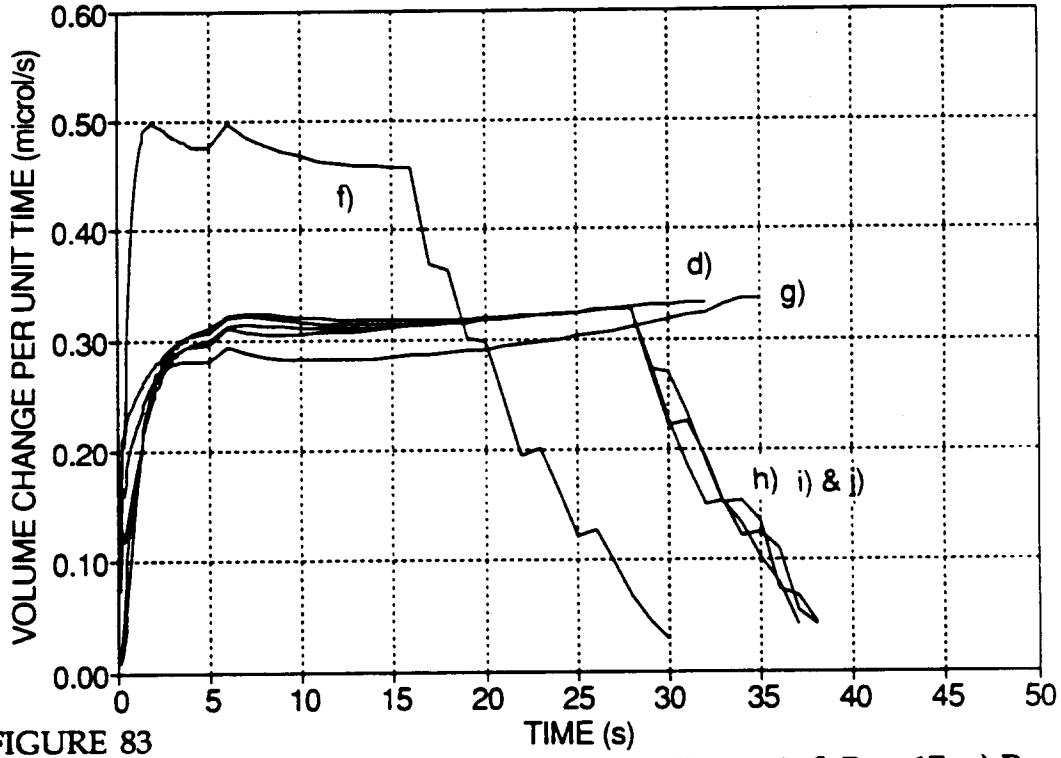

FIGURE 83
Transient Rate of Change of Droplet Volume: d) Run 5; f) Run 17; g) Run 20; h) Run 18; i) Run 16; j) Run 19 (Effect of Droplet Geometry)

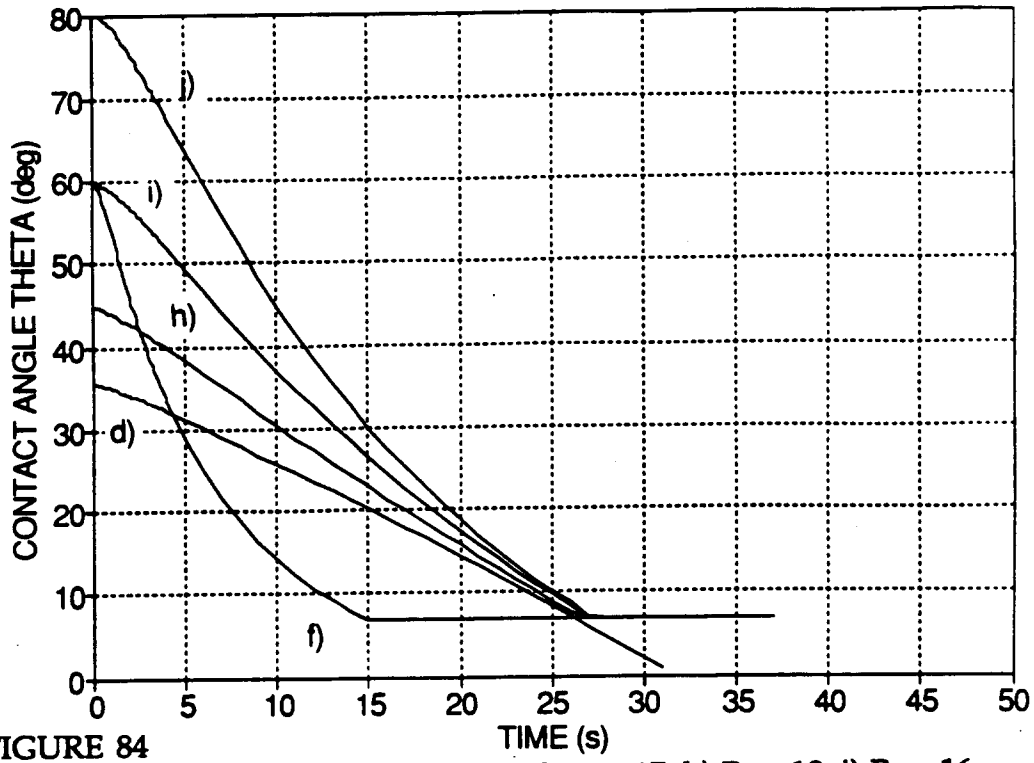

FIGURE 84
Transient Contact Angle θ: d) Run 5; f) Run 17; h) Run 18; i) Run 16; j) Run 19 (Effect of Droplet Geometry)

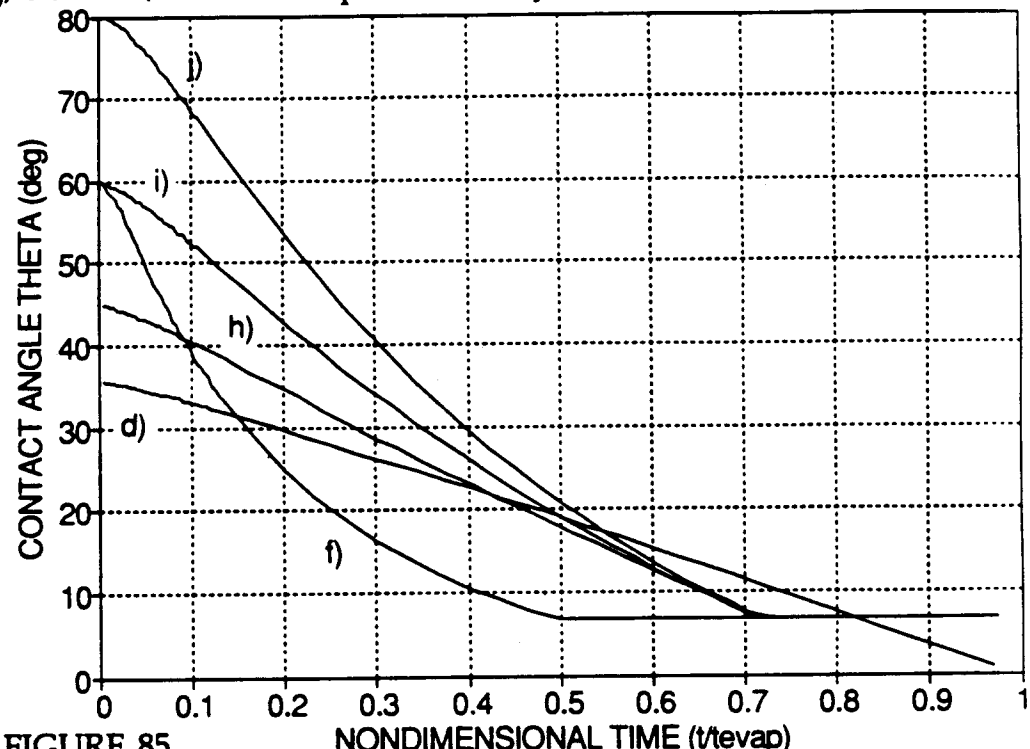

FIGURE 85
Contact Angle θ vs. Time Scaled by τ: d) Run 5; f) Run 17; h) Run 18; i) Run 16; j) Run 19 (Effect of Droplet Geometry)

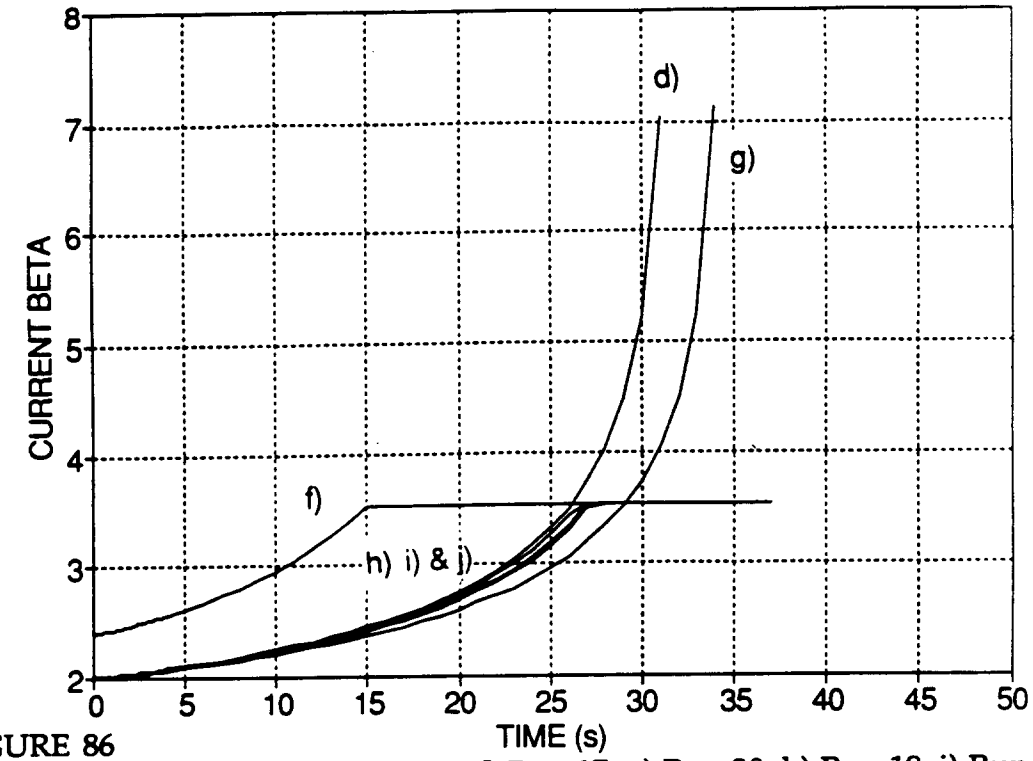

FIGURE 86
Transient Shape Factor β: d) Run 5; f) Run 17; g) Run 20; h) Run 18; i) Run 16; j) Run 19 (Effect of Droplet Geometry)

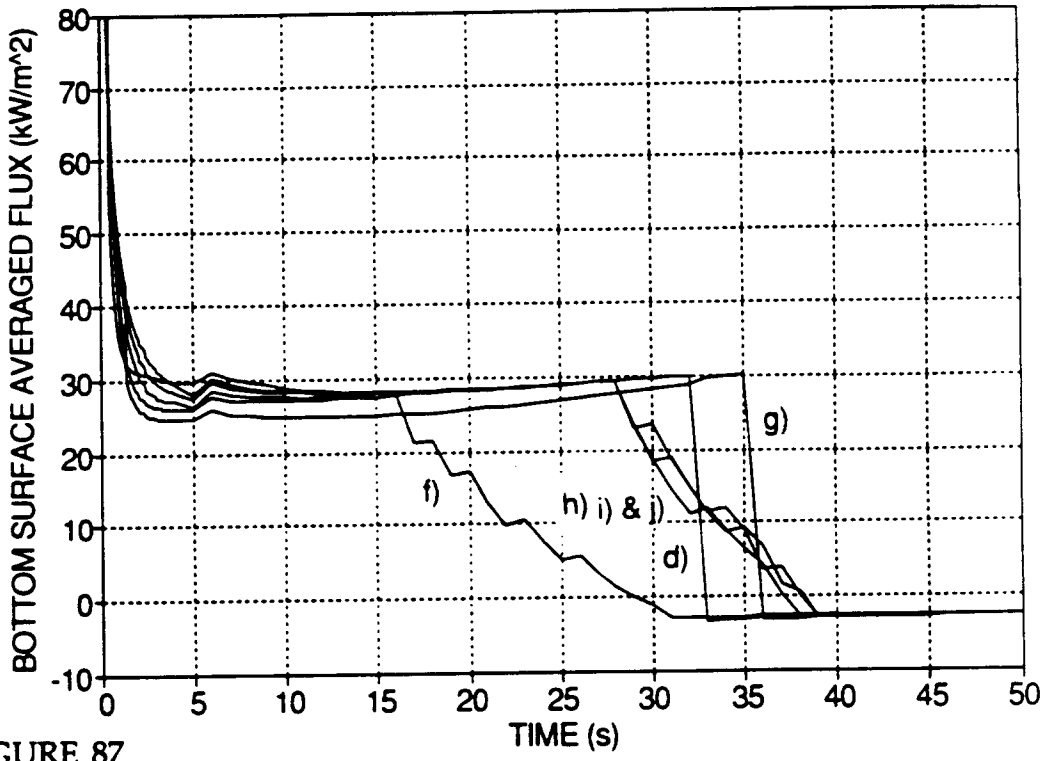

FIGURE 87
Transient Area Averaged Solid-Liquid Flux: d) Run 5; f) Run 17; g) Run 20; h) Run 18; i) Run 16; j) Run 19 (Effect of Droplet Geometry)

Appendix E FORTRAN Code for Constant Heat Flux Model

```
C     GLENN WHITE
C     (301) 405-5334
C     ADVISOR: MARINO DIMARZO  (301) 405-5257
C     MASTER'S THESIS -- UNIVERSITY OF MARYLAND AT COLLEGE PARK
C     EVAPORATIVE COOLING WITH RADIANT HEAT INPUT
C     WRITTEN FOR NIST CENTER FOR FIRE RESEARCH
C
C     SUDDEN PLACEMENT AND REMOVAL OF DISK HEAT SINK ON A SEMI-INFINITE SOLID
C
C234567
      PROGRAM DISK
      REAL RAD0, TIME(24), TEVAP, T, TS0, KS, ALPHA, R, QD, QI, ARG1, ARG2, ARG3
      INTEGER I, J
C.....RAD0: RADIUS OF WETTED AREA; TIME: TIME SINCE INTRODUCTION OF DISK
C.....TEVAP: EVAPORATION TIME; T: SURAFCE TEMPERATURE; TS0: INITIAL SURFACE TEMPERATURE;
C.....KS: CONDUCTIVITY OF SOLID; ALPHA: THERMAL DIFFUSIVITY OF SOLID;
C.....R: RADIAL POSITION; QD: DISK FLUX; QI: INITIAL FLUX;
C.....ARG1, ARG2, ARG3: ARGUMENTS TO SOLUTION INTEGRAL

C.....SPECIFY VARABLES TO BE USED BY SUB QAGI
      REAL EPSABS, EPSREL, RESULT, ABSERR, BOUND, WORK(5000), F
      INTEGER INF, IER, LIMIT, LENW, IWORK(1000), NEVAL, LAST
      EXTERNAL F
      COMMON ARG1, ARG2, ARG3
      OPEN (30, FILE = 'valdisk1.dat')

C.....DEFINE PHYSICAL PARAMETERS
      RAD0 = 2.673009E-3
      TS0 = 130.
      QD = 36534.
      QI = 2578.
      KS = 1.297
      ALPHA = 5.79E-7
      TEVAP = 31.5

C.....DEFINE THE TIMES TO BE USED
      DATA TIME /.1, .2, .3, .5, 1., 2., 3., 4., 5., 10., 15., 20., 25., 30.,
     +   31., 32., 33., 34., 35., 40., 45., 50., 55., 60./

C.....DEFINE VARIABLES NEEDED BY SUB QAGI (CMLIB LIBRARY) USED FOR CALCULATION OF SEMI-INFINITE INTEGRAL
      BOUND = .00001
      INF = 1
      EPSABS = .001
      EPSREL = .001
      LIMIT = 1000
      LENW = 5000

C.....DO LOOP FOR CALCULATING THE DIMENSIONLESS TEMPERATURE AS A FUNCTION OF R
      DO 15 I = 1, 24
        DO 10 J = 1, 78
          IF (J .LE. 48) THEN
            R = RAD0 * (J - .5) / 12.
           ELSEIF (J .LE. 66) THEN
            R = RAD0 * (4. + ((J - 48) - .5) / 6.)
           ELSEIF (J .LE. 75) THEN
            R = RAD0 * (7. + ((J - 66) - .5) / 3.)
           ELSE
            R = RAD0 * (10. + ((J - 75) - .5))
          END IF
          ARG1 = R / RAD0
          ARG2 = (ALPHA * TIME(I)) ** .5 / RAD0
          IF (TIME(I).LE.TEVAP) THEN
            ARG3 = 0.
           ELSE
            ARG3 = (ALPHA * (TIME(I) - TEVAP)) ** .5 / RAD0
          END IF
```

```
      CALL QAGI (F,BOUND,INF,EPSABS,EPSREL,RESULT,ABSERR,NEVAL,IER,LIMIT,LENW,LAST,IWORK,WORK)
      T = TS0 - (QD + QI) * RAD0 * RESULT / KS
      PRINT *, TIME(I), J, T
      WRITE (30, 5) TIME(I), J, R / RAD0, T
   10 CONTINUE
   15 CONTINUE
    5 FORMAT (F8.5, I5, 2F15.6)
      STOP
      END


      REAL FUNCTION F(X)
      REAL X
      REAL ARG1, ARG2, ARG3
      REAL ERF, BESI0E
C.....ERF: ERROR FUNCTION; BESJ0, BESJ1: BESSEL FUNCTIONS J0 AND J1
C.....(FROM LINKED LIBRARY)
      COMMON ARG1, ARG2, ARG3
      F = BESJ0(ARG1 * X) * BESJ1(X) * (ERF(ARG2 * X) - ERF(ARG3 * X)) / X
      RETURN
      END
```

References

1. Toda, S., "A Study of Mist Cooling. First Report: Investigation of Mist Cooling," *Heat Transfer, Japanese Research, 1:* 3, 39–50, 1972.

2. Bonacina, C., Del Giudice, S., and Comini, G., "Dropwise Evaporation," *Transactions ASME, Journal of Heat Transfer, 101:* 441–446, 1979.

3. Inada, S., Mikasaka, Y., and Nishida, K., "Transient Heat Transfer for a Water Drop Impinging on a Heated Surface," *Bulletin of JSME, 28:* 246, 2675–2681, 1985.

4. Makino, K. and Michiyoshi, I., "The Behavior of a Water Droplet on Heated Surfaces," *International Journal of Heat and Mass Transfer, 27:* 781–791, 1984.

5. diMarzo, M. and Trehan, A., "Transient Cooling of a Hot Surface by Droplets Evaporation," *National Institute of Standards and Technology Report,* NBS-GCR-86-516, 1986.

6. Klassen, M. and diMarzo, M., "Transient Cooling of a Hot Surface by Droplets Evaporation," *National Institute of Standards and Technology Report,* NIST-GCR-90-575, 1990.

7. Klassen, M., diMarzo, M., and Sirkis, J., "Infrared Thermography of Dropwise Evaporative Cooling," *ASME-HTD, 141:* 117–121, 1990.

8. Tartarini, P., Liao, Y., Kidder, C., and diMarzo, M., "Transient Cooling of a Hot Surface by Droplets Evaporation," *University of Maryland at College Park Mechanical Engineering Department Report,* No. 91-1, 1991.

9. diMarzo, M., Kidder, C., and Tartarini, P., "Infrared Thermography of Dropwise Evaporative Cooling of a Semi-Infinite Solid Subjected to Radiant Heat Input," *Experimental Heat Transfer, 5:* 101–104, 1992.

10. diMarzo, M., Kavoosi, F., and Klassen, M., "Transient Cooling of a Hot Surface by Droplets Evaporation," *National Institute of Standards and Technology Report,* NIST-GCR-89-559, 1989.

11. Kavoosi, F., diMarzo, M., Baum, H., and Evans, D., "An Application of Boundary Element Methods to a Transient Axisymmetric Heat Conduction Problem," *ASME-HTD, 110:* 79–85, 1989.

12. Liao, Y., "Dropwise Evaporative Cooling of Solid Surfaces," Doctoral Dissertation, University of Maryland at College Park, 1992.

13. Dawson, H. and diMarzo, M., "An Experimental Study of Multiple Droplet Evaporative Cooling," *University of Maryland at College Park Mechanical Engineering Department Report*, No. 92-1, 1992.

14. Tartarini, P., Liao, Y., and diMarzo, M., "Numerical Simulation of Multi-Droplet Evaporative Cooling," Presented at *The Tenth National Congress on Heat Transfer*, Universita degli Studi di Genova, Italy, 1992.

15. diMarzo, M., Tartarini, P., Liao, Y., Evans, D., and Baum, H., "Evaporative Cooling Due to a Gently Deposited Droplet," *International Journal of Heat and Mass Transfer*, in press, 1993.

16. Incropera, F. and DeWitt, D., *Fundamentals of Heat and Mass Transfer*, John Wiley and Sons, New York, 1990.

17. Viskanta, R. and Toor, J., "Absorption of Solar Radiation in Ponds," *Solar Energy, 21:* 17–25, 1978.

18. Brewster, M., *Thermal Radiative Transfer and Properties*, John Wiley and Sons, New York, 1992.

19. Hale, G. and Querry, M., "Optical Constants of Water in the 200-nm to 200-μm Wavelength Region," *Applied Optics, 12:* 3, 555–563, 1973.

20. Beck, J., Cole, K., Haji-Sheikh, A., and Litkouhi, B., *Heat Conduction Using Green's Functions*, Hemisphere Publishing Corporation, Washington, DC, 1992.

21. Personal Communication with M. diMarzo, 1992.

22. Al-Khafaji, A. and Tooley, J., *Numerical Methods in Engineering Practice*, Holt, Rinehart, and Winston, Philadelphia, 1986.

23. Keenan, J., Keyes, F., Hill, P., and Moore, J., *Steam Tables*, S.I. Ed., John Wiley and Sons, New York, 1978.

24. Chandra, S. and Avedisian, C., "On the Collision of a Droplet with a Solid Surface," *Proceedings of the Royal Society, 432:* 13–41, 1991.

25. Simon, F. and Hsu, Y., "Wetting Dynamics of Evaporating Drops on Various Surfaces," *NASA Tech. Memo.*, NASA TM X-67913, 1971.

26. Keenan, J., Chao, J., and Kaye, J., *Gas Tables*, Second Ed., John Wiley and Sons, New York, 1983.

27. Hilsenrath, J., *Tables of Thermodynamic and Transport Properties of Air, Argon, Carbon Dioxide, Carbon Monoxide, Hydrogen, Nitrogen, Oxygen, and Steam*, Pergamon Press, New York, 1960.

28. Bejan, A., *Advanced Engineering Thermodynamics*, John Wiley and Sons, New York, 1988.

29. Carslaw, H. and Jaeger, J., *Conduction of Heat in Solids*, Clarendon Press, Oxford, 1959.

NIST-114 (REV. 6-93) ADMAN 4.09

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

MANUSCRIPT REVIEW AND APPROVAL

(ERB USE ONLY)
ERB CONTROL NUMBER | DIVISION
PUBLICATION REPORT NUMBER: NIST-GCR-94-662 | CATEGORY CODE

INSTRUCTIONS: ATTACH ORIGINAL OF THIS FORM TO ONE (1) COPY OF MANUSCRIPT AND SEND TO THE SECRETARY, APPROPRIATE EDITORIAL REVIEW BOARD

PUBLICATION DATE: November 1994 | NUMBER PRINTED PAGES

TITLE AND SUBTITLE (CITE IN FULL)

Transient Cooling Of A Hot Surface By Droplets Evaporation

CONTRACT OR GRANT NUMBER: 70NANB1H1173

TYPE OF REPORT AND/OR PERIOD COVERED: Final Report, December 1993

AUTHOR(S) (LAST NAME, FIRST INITIAL, SECOND INITIAL)
G. White, S. Tinker, M. di Marzo
University of Maryland
College Park, MD 20742

PERFORMING ORGANIZATION (CHECK (X) ONE BOX)
[ ] NIST/GAITHERSBURG
[ ] NIST/BOULDER
[ ] JILA/BOULDER

LABORATORY AND DIVISION NAMES (FIRST NIST AUTHOR ONLY)

SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (STREET, CITY, STATE, ZIP)
U.S. Department of Commerce
National Institute of Standards and Technology
Gaithersburg, MD 20899

PROPOSED FOR NIST PUBLICATION

[ ] JOURNAL OF RESEARCH (NIST JRES)	[ ] MONOGRAPH (NIST MN)	[ ] LETTER CIRCULAR
[ ] J. PHYS. & CHEM. REF. DATA (JPCRD)	[ ] NATL. STD. REF. DATA SERIES (NIST NSRDS)	[ ] BUILDING SCIENCE SERIES
[ ] HANDBOOK (NIST HB)	[ ] FEDERAL INF. PROCESS. STDS. (NIST FIPS)	[ ] PRODUCT STANDARDS
[ ] SPECIAL PUBLICATION (NIST SP)	[ ] LIST OF PUBLICATIONS (NIST LP)	[ ] OTHER
[ ] TECHNICAL NOTE (NIST TN)	[ ] NIST INTERAGENCY/INTERNAL REPORT (NISTIR)	

PROPOSED FOR NON-NIST PUBLICATION (CITE FULLY) [ ] U.S. [ ] FOREIGN

PUBLISHING MEDIUM
[ ] PAPER [ ] CD-ROM
[ ] DISKETTE (SPECIFY)
[ ] OTHER (SPECIFY)

SUPPLEMENTARY NOTES

ABSTRACT (A 2000-CHARACTER OR LESS FACTUAL SUMMARY OF MOST SIGNIFICANT INFORMATION. IF DOCUMENT INCLUDES A SIGNIFICANT BIBLIOGRAPHY OR LITERATURE SURVEY, CITE IT HERE. SPELL OUT ACRONYMS ON FIRST REFERENCE.) (CONTINUE ON SEPARATE PAGE, IF NECESSARY.)

A computer code is developed and tested which simulates the transient evaporation of a single liquid droplet from the surface of a semi-infinite solid subject to radiant heat input from above. For relatively low temperature incident radiation, it is shown that the direct absorption of radiant energy by the droplet can be treated as purely boundary conditions, while a model for higher temperature incident radiation would require the addition of constant heat source terms. The heat equation is numerically coupled between the liquid and solid domains by using a predictor-corrector scheme. Three one-dimensional solution schemes are used within the droplet: a start-up semi-infinite medium solution, a tridiagonal Crank-Nicholson transient solution, and a steady-state solution. The solid surface temperatures at each time step are calculated through careful numerical integration of an axisymmetric Green's functions solution with the forcing function given by the past lower droplet surface and solid-vapor boundary heat fluxes. The time step in increased after a sensitive initial period to allow for reasonable run times. Two geometry models are include which give the droplet height as a function of current droplet volume and initial wetted radius; the second allows inclusion of the effects of initial contact angle and receding angle. Using water as the liquid and Macor, a low-thermal conductivity material, as the solid, the program output was compared to the experimental results in this line of research. They correlate well to the experiments in which the critical geometric shape factor and evaporation time were most easily measured.

KEY WORDS (MAXIMUM OF 9; 28 CHARACTERS AND SPACES EACH; SEPARATE WITH SEMICOLONS; ALPHABETIC ORDER; CAPITALIZE ONLY PROPER NAMES)
computer programs; droplets; evaporation; fire research; steady state; water

AVAILABILITY
[X] UNLIMITED [ ] FOR OFFICIAL DISTRIBUTION - DO NOT RELEASE TO NTIS
[ ] ORDER FROM SUPERINTENDENT OF DOCUMENTS, U.S. GPO, WASHINGTON, DC 20402
[X] ORDER FROM NTIS, SPRINGFIELD, VA 22161

NOTE TO AUTHOR(S): IF YOU DO NOT WISH THIS MANUSCRIPT ANNOUNCED BEFORE PUBLICATION, PLEASE CHECK HERE. [ ]

WORDPERFECT